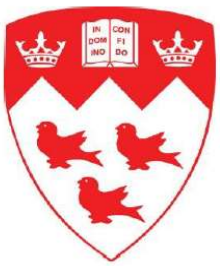
# McGill UNIVERSITY

845 Sherbrooke St W, Montreal,

QC H3A 0G4

## DEPARTMENT OF COMPUTER SCIENCE

### COMP 551- APPLIED MACHINE LEARNING

# MINIPROJECT 1: MACHINE LEARNING 101

**Submitted to - PROF. WILLIAM L. HAMILTON**

**Submission Date: 31-01-2019**


**Submitted by:**

**MOHAMMAD HAMED AZIZI - 260812541**

**SURYA KUMAR DEVARAJAN - 260815492**

**SAEED SHOARAYE NEJATI - 260890049**

## ABSTRACT

In this project we're going to apply machine learning algorithms on a dataset that is provided by reddit.com. The datasets have different features such as children, controversiality, is_root and text. Our task is to figure out a model that will predict the popularity of a comment. we'll start by defining linear regression algorithms-closed form and gradient descent and compare the results by implementing them, considering different features and different parameters for each task. Our project is divided into three tasks: first, we will split our dataset into training, validation and test datasets and extract the desired features for every task. Then, we'll be implementing the closed form and gradient descent algorithms. Finally, we'll compare the results of these algorithms on the validation set to check the performance and stability of our models. Getting the best model that we have and running it on the test set will examine our trained model when it comes to unseen dataset. We found that the gradient descent approach was slower than the closed-form approach for the dataset provided and we analysed how decay plays a role in gradient descent.


## INTRODUCTION

This project involves the use of the Linear Regression models to predict the popularity of the comments on Reddit. We are provided with a large set of data of reddit comments which also has features like children, Controversiality and is_root. children. We also have the information on the popularity score. It measures the popularity of the comments and it is the target variable that we need to predict.

Machine learning, more specifically the field of predictive modeling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible. For example, in a simple regression problem (a single **x** vector and a single y), the form of the model would be:

$$y = w_0 + \sum_{i=1:n} w_i \boldsymbol{x}_i$$

From least square solution:

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

Gradient descent is an optimization algorithm that uses the decay to control the step size to minimizes the loss. Decay here depends on the value of η and β. In this project, we have showed the results of using both algorithms and comapred them for getting the best algorithm for our dataset.

For our given dataset, changing the number of features in each experiment will show us which model(closed form and gradient descent) fitsthe data. Evenmore, changing the parameters of gradient descent will have a big impact on the stability, runtime and performance of the model.

In the next part of this project we will show how we implemented these two algorithms and apply them taking different features in each task. We referred to "Jason Brownlee (2017, Oct 20), How

to Develop a Deep Learning Bag-of-Words Model for Predicting Movie Review Sentiment. Retrieved from https://machinelearningmastery.com/deep-learning-bag-of-words-model-sentiment-analysis/"

# Dataset

The first part of the project is to split the data sets into Training, Validation and Testing sets. We use first 10000 data points for training, next 1000 for validation and the last 1000 as test sets. The features other than text does not need preprocessing as we can use the numerical values and binary values as it. Children feature counts the number of replies a comment received. Controversiality feature is a metric of controversial of a comment and takes on the binary values. is_root feature is a binary variable that indicates if a comment is the root comment of a discussion thread. The preprocessing of text involves two operations, making all the text to lower case and split the text according to white spaces to define different words. After preprocessing, we are extracting 160 most frequently occurring words over all the comments that we have taken in the training set. Every comment will have a 160-dimensional feature vector. The word count feature is done by creating a bag of 160 most frequently occurring words and compare this to the vector such that the frequency of the most occurring words is filled in that vector.

Moreover, generating a model with more accuracy depends on the precision of the features. **To get more precise features we tried to clean the text feature, by removing some words of prepositions, pronouns, articles.** for example; a dataset with text feature "I went to school" will be filtered to "went school". The top 160 frequent words before cleaning and after cleaning the dataset are attached with the report.

We come up with new feature which is the transformation of children feature, since the number of comments will have a great impact on the popularity. We will start squaring the children feature (MSE_closed_square) and compute the new value of MSE. then we will compute the MSE for closed form after adding the exp of children feature (MSE_closed_f2).

The main concern that might arise when working with public social media is that we are not dealing with a small dataset anymore, in real life we are getting millions of comments and replies on daily basis in a streaming continuous way, preprocessing a large streaming dataset will be an issue. In a way or another, this will push us to use another technique to deal with this kind of dataset.

# RESULTS

### Task 1

After implementing the two algorithms, we start by taking into consideration the first three features (children, is_root, controvasilty), and compare the performance and accuracy of these algorithms by calculating the runtime and the value of errors.

**Gradient descent**

In the below table, we are changing the value of η and β to get different decay learning values, then we calculate **the runtime**, loss and steps for each pair values of η and β.

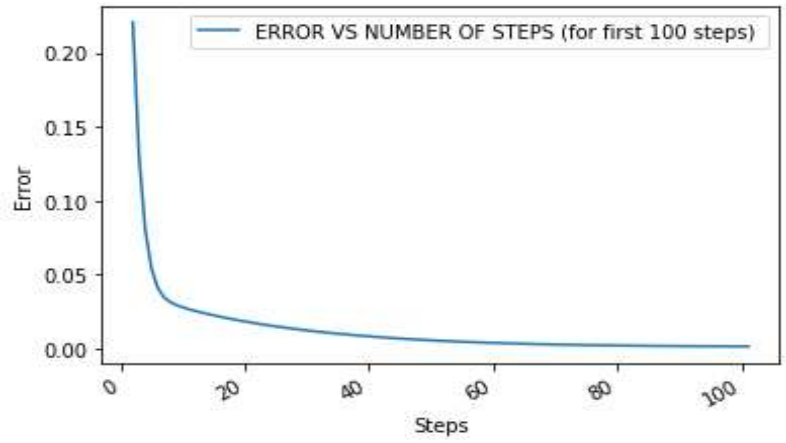
**Table 1**

| β | η | Runtime | Steps |
|---|---|---|---|
| 10e-3 | 10e-2 | 0.01847 | 117 |
| 10e-3 | 10e-3 | 0.018992 | 175 |
| 10e-3 | 10e-4 | It takes a long time | It takes a long time |
| **10e-3** | **10e-5** | **1.800957918** | **36333** |
| 10e-3 | 10e-6 | It takes a long time | It takes a long time |
| 10e-3 | 10e-7 | It takes a long time | It takes a long time |
| 10e-4 | 10e-2 | 0.04224681 | 118 |
| 10e-4 | 10e-3 | 0.0176391 | 190 |
| 10e-4 | 10e-4 | 0.0517199 | 608 |
| **10e-4** | **10e-5** | **0.3444151** | **6604** |
| **10e-4** | **10e-6** | **9.12999901** | **186161** |
| 10e-5 | 10e-2 | 0.0242769 | 118 |
| 10e-5 | 10e-3 | 0.0203499 | 190 |
| 10e-5 | 10e-4 | 0.0482759 | 590 |
| 10e-5 | 10e-5 | 0.2966492 | 5581 |
| **10e-5** | **10e-6** | **2.437783** | **47315** |

As we can see from the above table, for 10e-4 and 10e-5 pair of η and beta we're having the best performance with 9.9914e-9 error and 0.3444151 runtime.

***Mean square error = 1.34161955***

The **Stability** of closed form does not depend on any hyperparameters. It is straightforward calculation. But, the stability of Gradient descent mainly depends on the learning rate value which in turn depends on the hyperparameters eta and beta. When the learning rate is too big, it will not reach the local minimum because it just bounces back and forth between the convex function of gradient descent. If learning rate is very small, gradient descent will eventually reach the local minimum, but it will take too much time. Please refer to table 1 to see the number of steps and runtime taken for different values of η and β.




Comparing the runtime and stability (for values mentioned above β=10e-5 and η=10e-4) of gradient descent and closed form algorithms, we can conclude that the closed form has less MSE (**better performance**) and runtime than the gradient descent method.

**Table 2**

| TRAINING | VALIDATION |
|---|---|
| **Closed Form**:<br>*MSE= 1.084674*<br>*Runtime= 0.214557* | *MSE=1.011749*<br>*Runtime= 0.2057149* |
| **Gradient Descent**:<br>*MSE=1.34161955*<br>*Runtime= 0.3444151* | *MSE=1.27693395*<br>*Runtime= 18.5808629* |

**N.B: we're taking the best values of η and β that we got in Table 1**

From the above results, we can notice that the runtime of gradient descent is more since we are calculating the decay learning rate for each step. Moreover, the closed have a better accuracy since the loss is less than in gradient descent.

**Task 2**

Below are the results of closed form algorithm.

**Table 3**

| TRAINING | VALIDATION |
|---|---|
| **NO TEXT**:<br>*MSE= 1.084674*<br>*Runtime= 0.214557* | *MSE=1.011749*<br>*Runtime= 0.2057149* |
| **60 words**:<br>*MSE=1.061161*<br>*Runtime= 0.275542* | *MSE=0.904507*<br>*Runtime= 0.26471018* |
| **160 words**:<br>*MSE=1.0467629*<br>*Runtime= 0.284834* | *MSE=0.912364*<br>*Runtime=0.365887* |

For all features, no text feature and 60 frequent word feature, MSE on validation is slightly better than the training set. It is not entirely possible all the time but here we have values on validation due to some random noise. In these cases, the model neither overfits nor underfits.

**Task 3**

Below are the values of **MSE for closed form** applied on training set after adding the two new features;

| | |
|---|---|
| Closed_without new feature | 1.0467629 |
| Closed_ with square feature | 1.0005977 |
| Closed_ with exp feature | 1.0366922 |
| Closed with both new feature | 1.0003905 |

**We can see that the performance of the model improved by more than 4.6% after adding the two new features.**

**Task 4:**

Now we will run our model after adding the two new features on the test set to see how generalise is the model and how accurate it is for an unseen dataset.

Below are the values of **MSE for closed form** applied on training, validation and test set after adding the two new features;

**Table 5**

| | |
|---|---|
| Closed_training | 1.0003905 |
| Closed_validation | 0.756792 |
| Closed_test | 1.0185782 |

**Conclusion**

After running these two algorithms, we observed that closed form approach might have better performance for some datasets, and for a big dataset, gradient descent will take more time. Moreover, adding more features to our dataset will improve the performance of our model. These new features have great impact on training but lower on validation and test sets. MSE on training set, decrease on validation and test set because of data size (noise of small dataset).

**Contribution**

We divided the work load that involves analysing the tasks, coding, running, testing and writing the report equally. We gathered once in every 2-3 days, working together towards the completeness of the project.