CSCI B555 Machine Learning

Programming Project 3

Bayesian Generalized Linear Models

Introduction

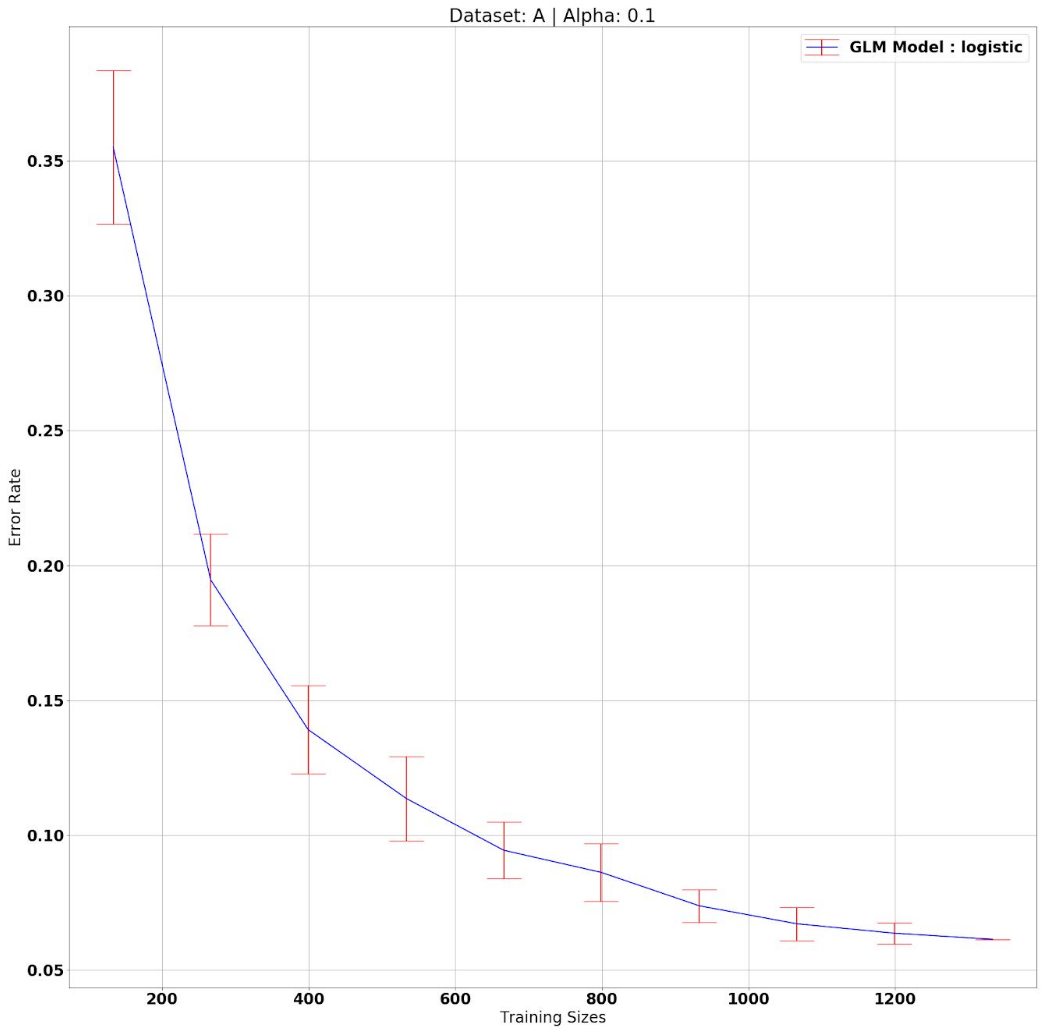
The goal of this project is to develop one generic implementation for the main GLM algorithm that works for multiple observation likelihoods for Logistic, Poisson and Ordinal Regression for Classification. Newton Raphson method is used to estimate *w*Map which is used for prediction.

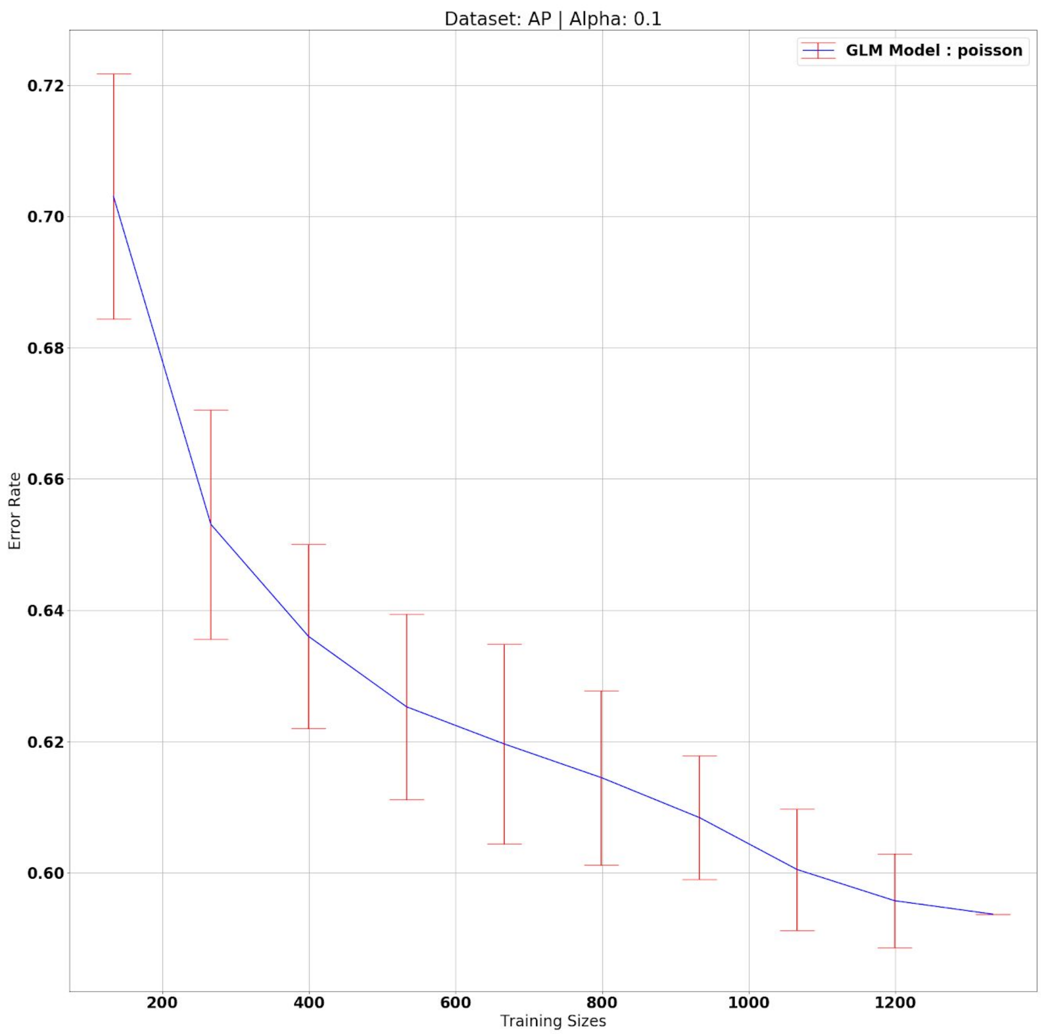
The code implements a generic function of GLM variant that has access to procedures that calculate first and second derivatives based on the model. *w*Map is then calculated using Newton Raphson method. We use 2/3rd of the dataset for training purposes. Using the value of *w*Map obtained, we predict the classification labels of the test set and compare the results with the True labels to calculate the error based on misclassified labels.

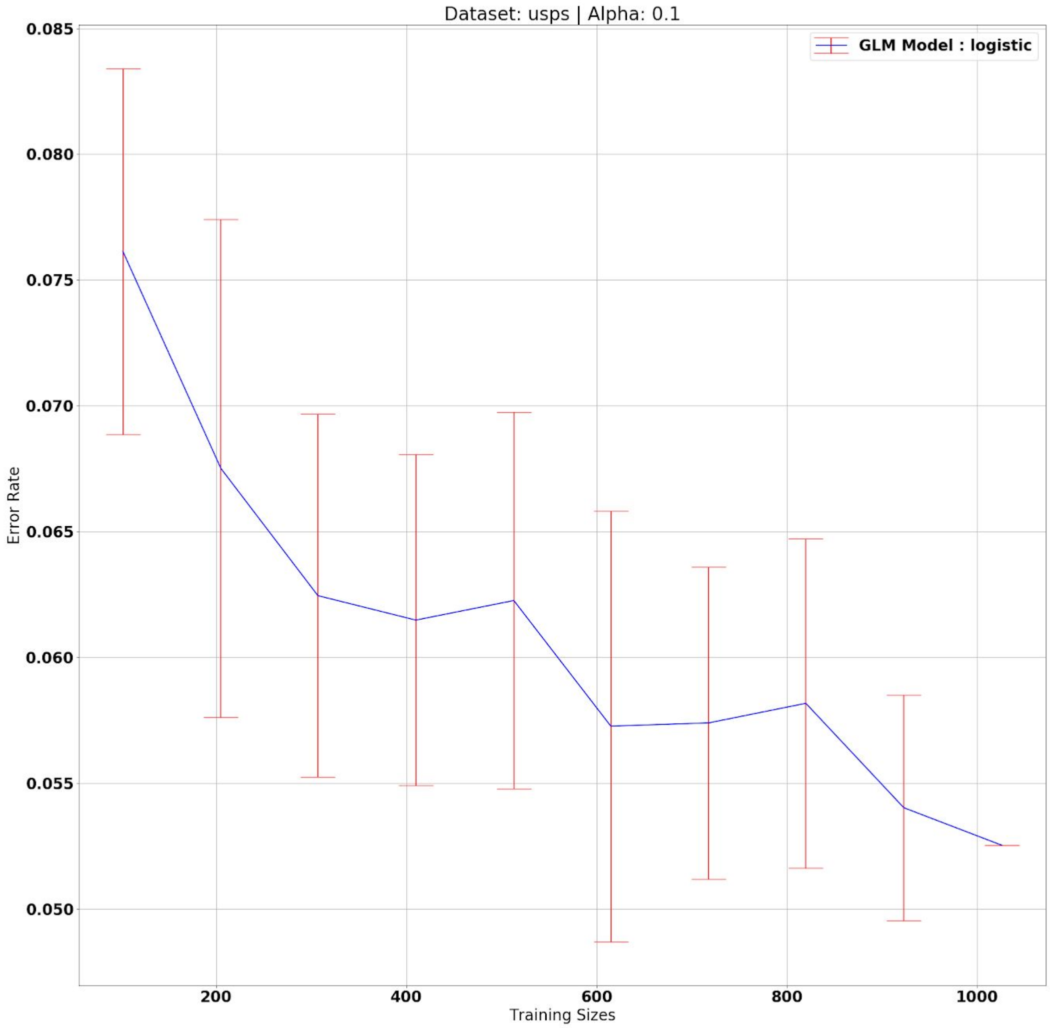
We run this iteratively on increasing portions of training data set sizes - 0.1, 0.2.... 1 and find the mean and standard deviation of error from 30 trials.

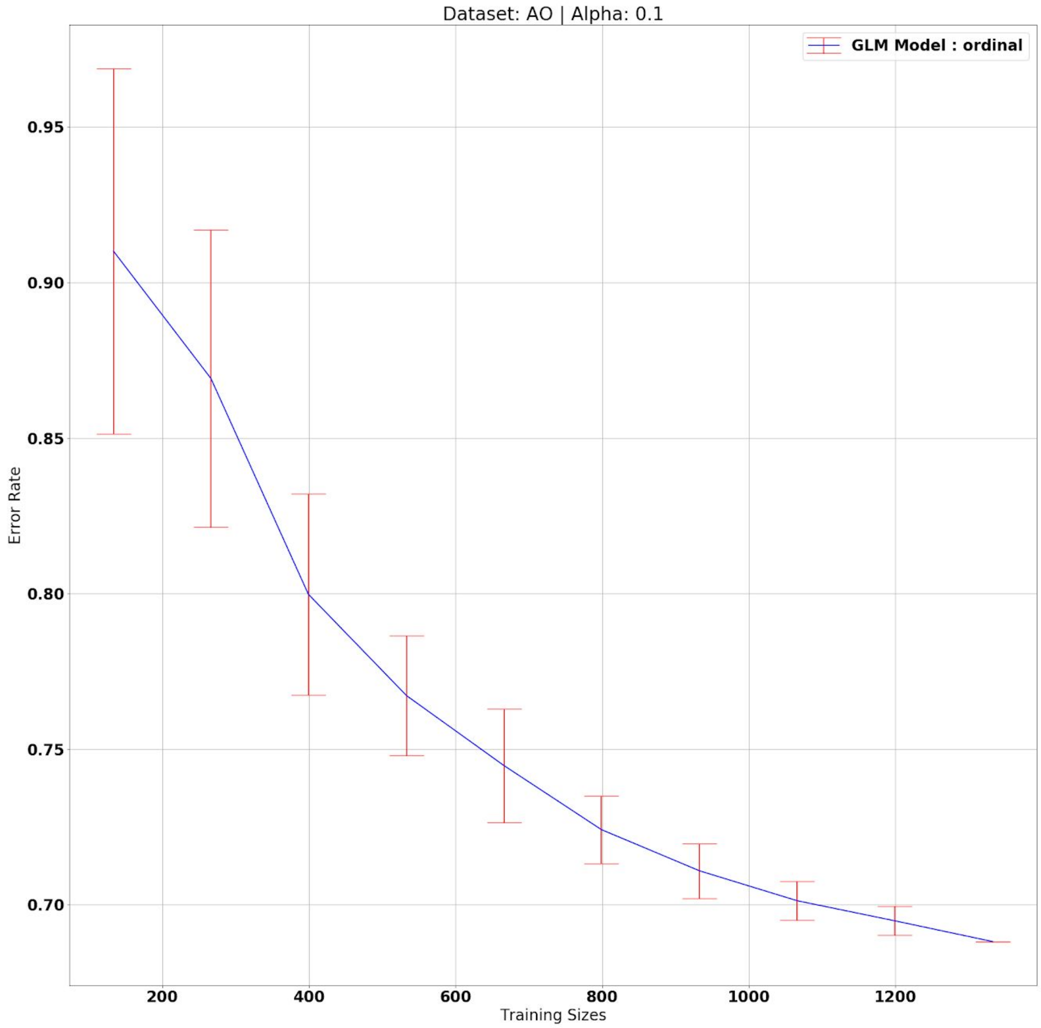
Following are the plots for error rates vs training set sizes:

1) Alpha = 0.1

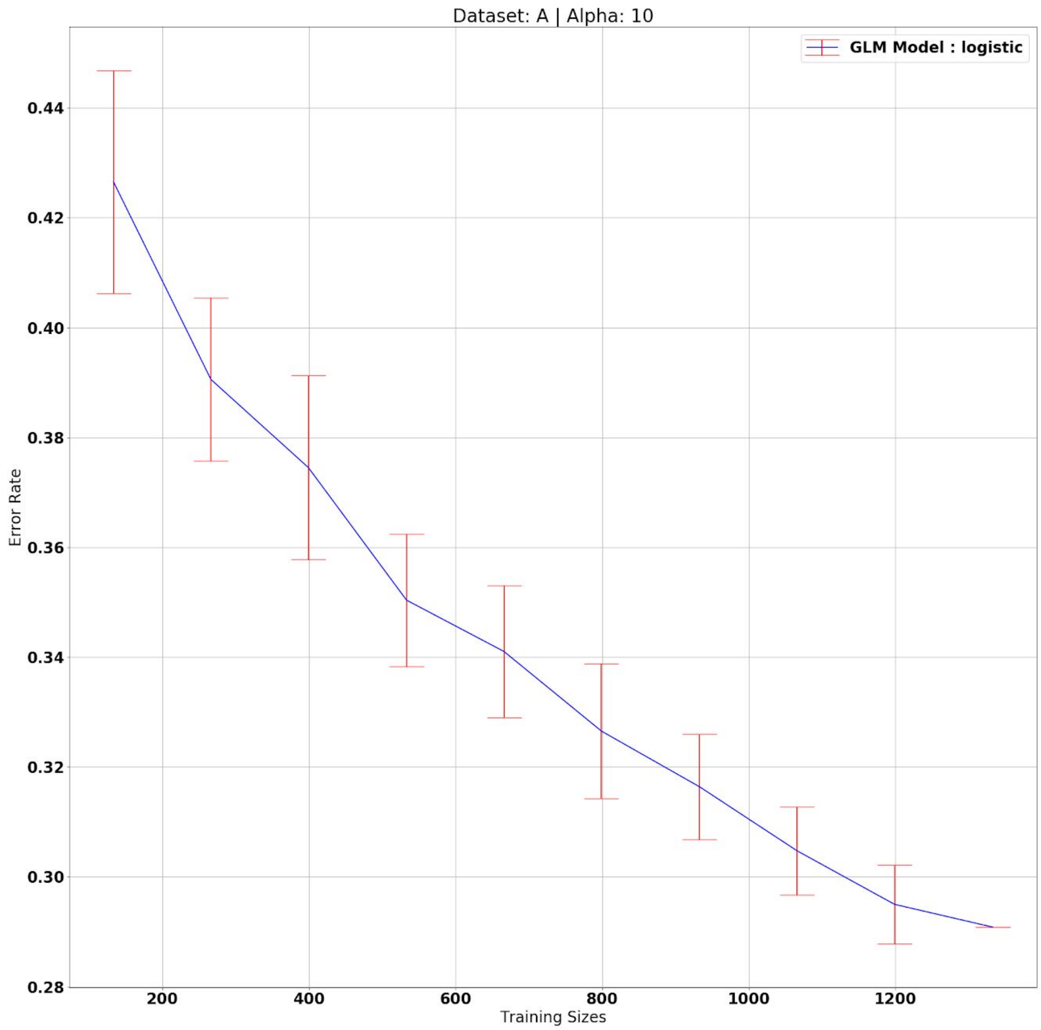

Dataset: A | Alpha: 0.1
GLM Model : logistic
Error Rate
0.35
0.30
0.25
0.20
0.15
0.10
0.05
200
400
600
800
1000
1200
Training Sizes

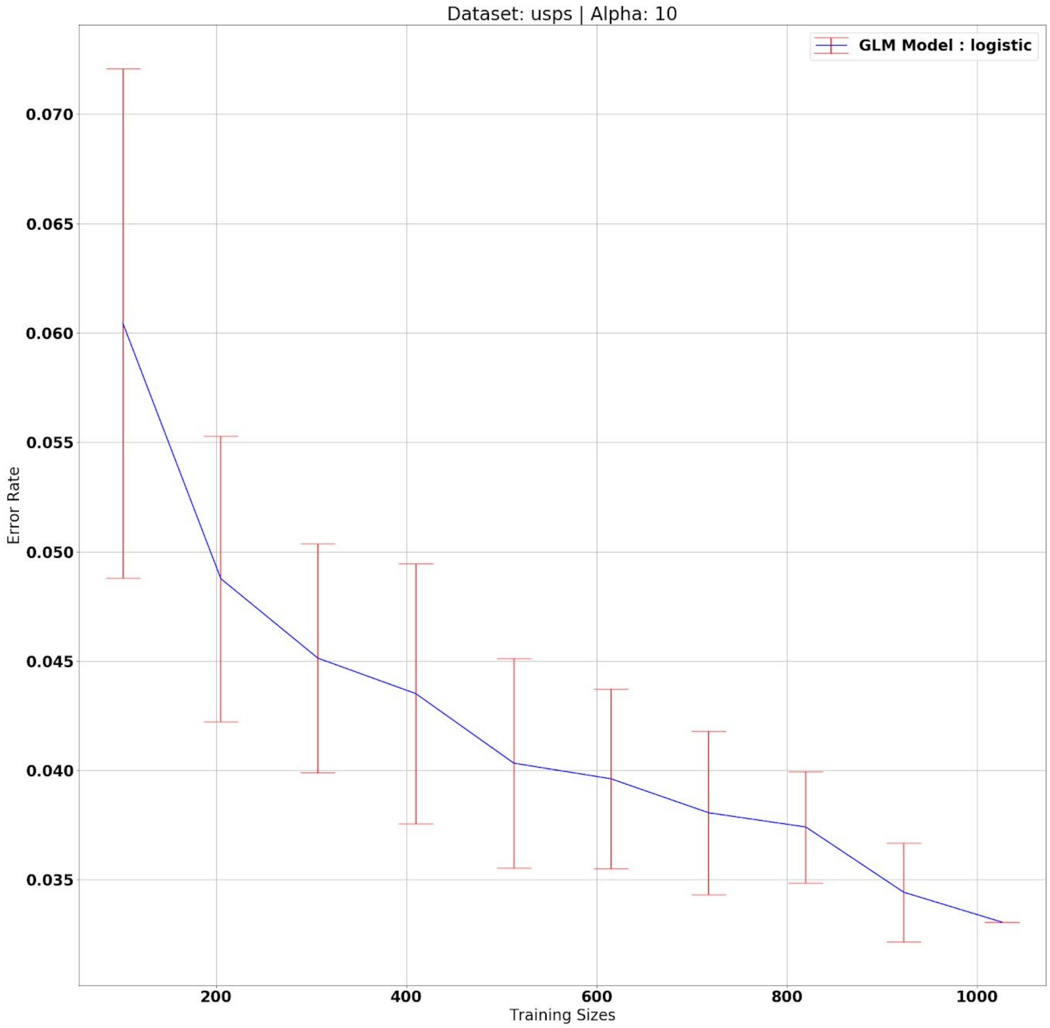

Dataset: AP | Alpha: 0.1
GLM Model : poisson
Error Rate
0.72
0.70
0.68
0.66
0.64
0.62
0.60
200
400
600
800
1000
1200
Training Sizes

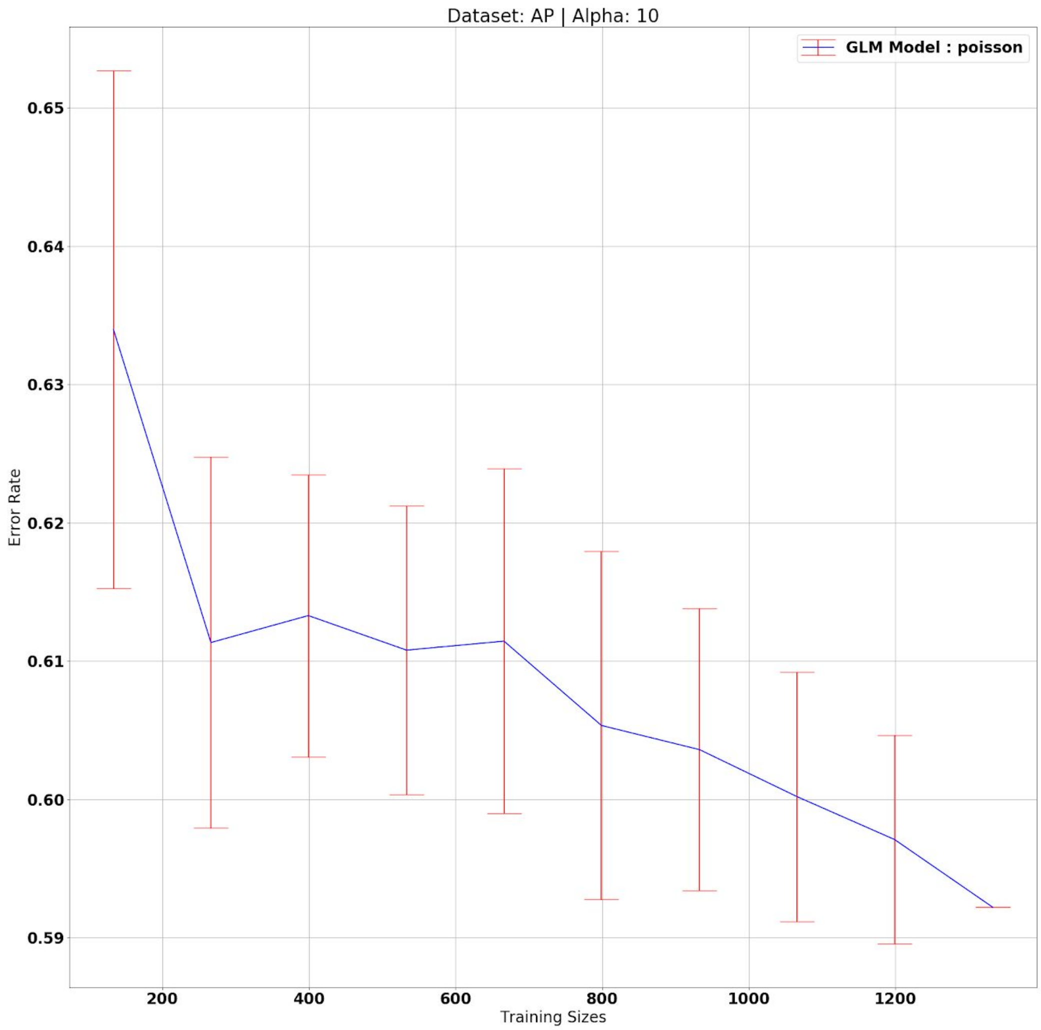

Dataset: usps | Alpha: 0.1
GLM Model : logistic
Error Rate
0.085
0.080
0.075
0.070
0.065
0.060
0.055
0.050
200
400
600
800
1000
Training Sizes


Dataset: AO | Alpha: 0.1
GLM Model : ordinal
0.95
0.90
0.85
0.80
0.75
0.70
Error Rate
200
400
600
800
1000
1200
Training Sizes

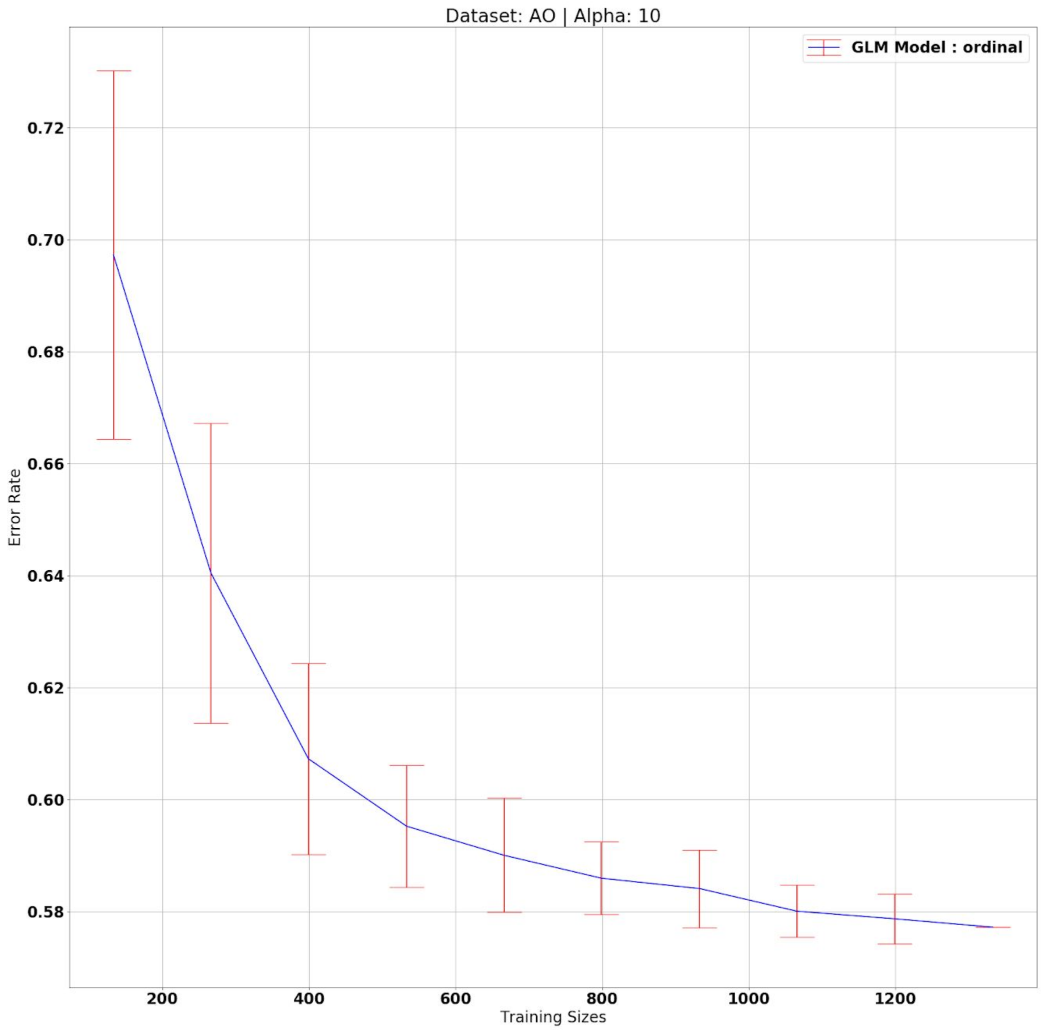
2) Alpha = 10


Dataset: A | Alpha: 10
GLM Model : logistic
Error Rate
Training Sizes
0.44
0.42
0.40
0.38
0.36
0.34
0.32
0.30
0.28
200
400
600
800
1000
1200


Dataset: usps | Alpha: 10
GLM Model : logistic
Error Rate
Training Sizes
0.070
0.065
0.060
0.055
0.050
0.045
0.040
0.035
200
400
600
800
1000



Dataset: AP | Alpha: 10
GLM Model : poisson
0.65
0.64
0.63
0.62
0.61
0.60
0.59
Error Rate
200
400
600
800
1000
1200
Training Sizes



Dataset: AO | Alpha: 10
GLM Model : ordinal
Error Rate
Training Sizes
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
200
400
600
800
1000
1200

Results and Discussion

(Output statistics are printed in folders alpha0_1 and alpha10, as files - 'Output Summary <model name> <dataset name>.txt'

- As expected, the error rate reduces with increasing dataset sizes. The model classifies better when it is trained with more data. Hence, the learning curve is as expected.
- With increasing dataset size, the error is also much less deviated from the mean. As expected we get a better performance from the classifier when trained with larger dataset.
- For logistic regression model, datasets A(2000 X 60) and usps (1540 X 256) were tested. We expect W calculation to take longer for more features. The results show a similar pattern, the model runs much faster for A and computes *w*Map in about 4 (6-7 for $\alpha = 0.1$) iterations and takes 7-8 iterations (10-11 for $\alpha = 0.1$) to converge for usps dataset.
- For poisson we used AP dataset (2000 X 60) and for Ordinal we use AO(2000 X 60) sample dataset. For logistic we want to predict labels = {0,1}, for poisson we perform count prediction, and for ordinal we predict labels = {1,2,3,4,5}. We expect a better performance for logistic since there are just 2 classes to predict. Mean performance based on error is best for logistic model. Ordinal performs slightly better than Poisson. Mean Error is much deviated for Poisson which is intuitive since we have more classes to predict.
- Model Performance Comparison based on Run time and No. of iterations : Logistic Model performs the best and runs the fastest. Ordinal takes more time due to more computations but computes W in fewer iterations as compared to Poisson given that all 3 models have the same number of features and samples.

Model Selection

Based on our last programming project, it looked like a good idea to extend Cross Validation to Linear models for classification as well. I used 10 fold cross validation here. The model is trained using the training set portions of the dataset setting aside the Kth portion. After finding *w*Map, we use the Kth portion to test the model and find error rate. This is done for parameters ranging from 0.1,0.2...0.9,1,2...100. The best parameter is chosen from the minimum error rate achieved.

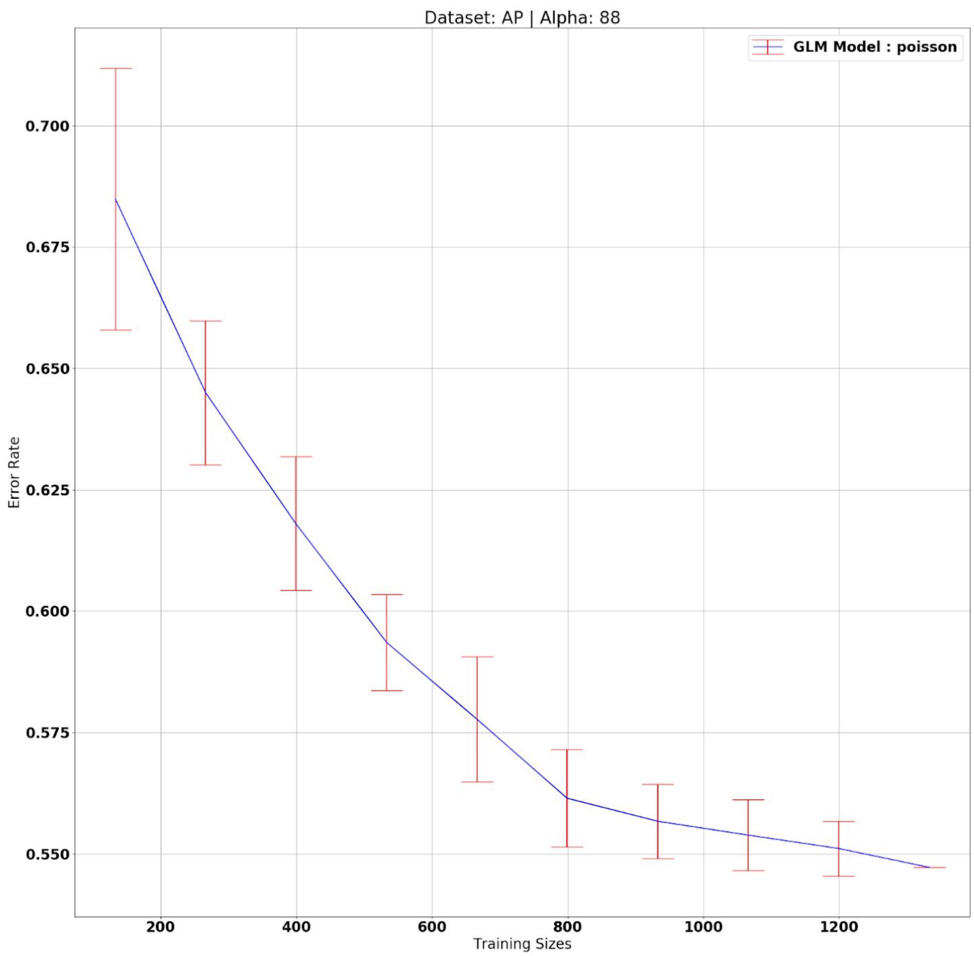
The results obtained below show that it works well for logistic regression which as seen from before gives best results for α = 0.1.

```
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py A logistic
Dataset:  A
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  0.1
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py A logistic
Dataset:  A
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  0.1
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py A logistic
Dataset:  A
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  0.1
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3>
```

For poisson regression, an average α = 88 was obtained.

```
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  83
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  96
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  92
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  100
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  80
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AP poisson
Dataset:  AP
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  82
..Done!
```

I tested my code to plot the error rate when α = 88. Following plot was obtained:



This is showing better results when compared to lower α values of α = 0.1 and α = 10.

For Ordinal, we get α = 3.

```
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3> python .\model_selection.py AO ordinal
Dataset:  AO
--MODEL SELECTION USING CROSS VALIDATION--
Parameter:  3
..Done!
PS C:\Neha\Grad Sem 1\Machine Learning\Assignments\Assignment 3>
```