Handwriting Recognition in Forensics

Pratik Pravin Kubal
SUNY,University At Buffalo
pkubal@buffalo.edu

1 Introduction

We are going to compare three different models of Linear Regression, Logistic Regression and Neural Network. We find that Neural network gives highest accuracy while training on forensic data. We also find that pixel value datasets such as GSC are works best in Neural network settings. We also explore some methods for Dataprocessing and Linear regression.

2 Methods

2.1 Data Processing

For Linear Regression we first cluster the data around the centroids, and use a representative clusters[1] such that each instance of the data is included in the train,test, and validation set. We use Elbow Method[1],discussed in 2.2 to determine the optimal number of clusters.
For Logistic Regression we use a stratified sampling method based on target classes to determine the train,test and validation set.
While using neural network the model uses a k fold technique on validatation set therefore we partition the dataset based on startified sampling into Train and test, Where train will be further divided into Train and Validation.Refer table
Optionally we could also Normalize the data based on the Standard deviation and mean of the data, but we didn't find any differences after Normalizing the data during Linear regression.

2.2 Elbow Method to find Optimal Clusters in Linear Regression

Elbow method computes the Sum of Squared Distances for a range of clusters, we are use the range from 2 to 15 because of the computational complexity of running the covariance, and Design Matrix constructions for large valuues of k. To find the optimal cluster number we have to draw a line connecting the first cluster value to the last, and based on the distance of points(cluster points) we have to select the one with largest distance, which will give us the Elbow of the graph. Refer to Fig.1,2,4 for Graphs for Human Observed with feature concatination and substraction and GSC dataset with feature concatination and substraction respectively. The cluster sizes we are going to use are given in Table 1.

2.3 Unbalanced Data

We have more instances of negative instances than positive matches. This is evident in datasets such as Credit Card Fraud, Fraud in General. There are various methods to deal with unbalanced data[2][3]. We are constrained by the project defination and it is not possible to generate new data. Furthermore, we are more concerned with finding positive instances, since these positive instances are highly relevant to the problem defination. Moreover, classifying a positive instance as negative(Which can

[1] Refer to my previous report on Project 1.2

Table 1: Optimal Cluster Sizes For Datasets

Model	Cluster Size
Human Observed Feature Concatenation	5
Human Observed Feature Substraction	5
GSC Feature Concatenation	5-6
GSC Feature Substraction	5

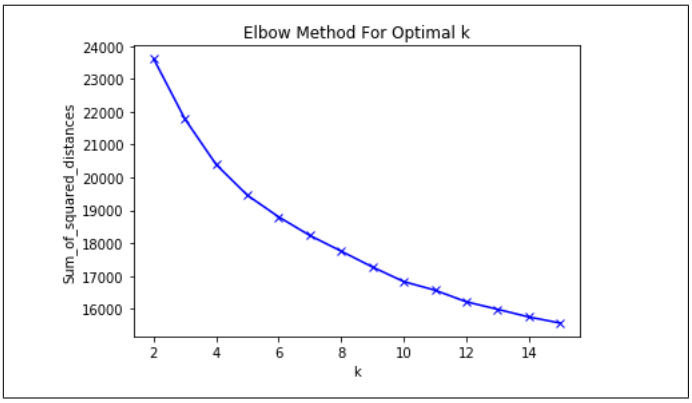

Figure 1: Human Observered Feature Concatenation

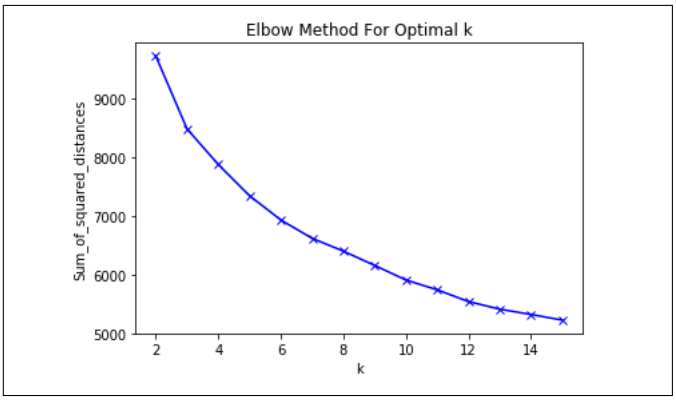

Figure 2: Human Observed Feature Substraction

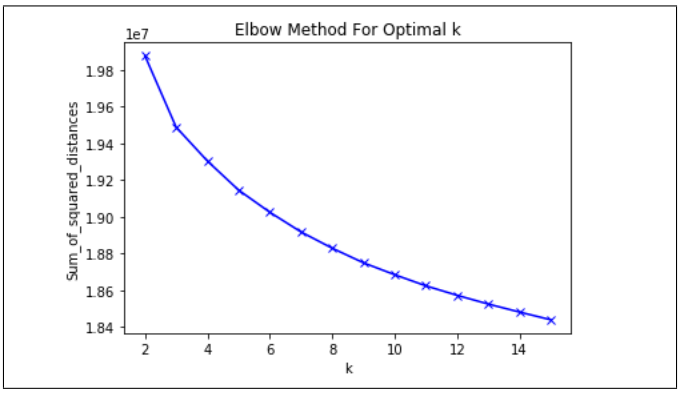

Figure 3: GSC Feature Concatenation

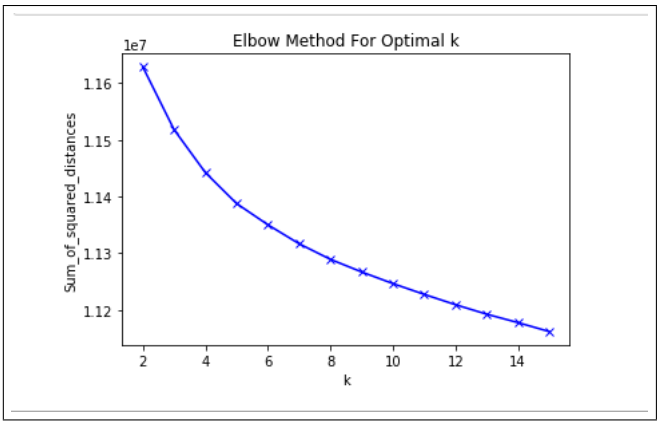

Figure 4: GSC Feature Substraction

be done by predicting all instances as 0 and getting a 99% accuracy) is bad than to classify a negative instance as positive.
We use two techniques in Linear and Logistic Regression for sampling, viz. Perfect sampling, where we take equal amount of postive and negative samples, and Over Sampling where we take more instances of Positive than negative.
We need to adress that while taking more instances of Positive targets we could overtrain the model, therefore we need to be carefull about this during Experimentation Phase.

2.4 Epoch Shuffle

Following more classical defination of SGD[4] we shuffle the training and the design matrix preserving the order or index in both. Our basic approach for SGD using Epoch shuffle will be:

1. Intiate Random or Zero weights
2. While change in loss does not equal to less than the sensitivity defined
2.a for each n=1...len(train)
2.a.i Compute change in weights
2.b Shuffle the train and Design matrix
2.c Calculate the change in loss.

While figuring out the optimal hyper parameters we also include if we want to use epoch shuffling or not.

3 Experimentation

3.1 Linear Regression

For Linear regression we are using a non uniform Gaussian Distribution. Here the variance increases as the input features increase. In past we have found good results using this Gaussian Distribution.
Instead of logging the E_{rms} values for all the iteration we log them at epoch values, this makes the code run faster. Since SGD in different epoch should still minimize the error, we will find that this method will not fail to find optimal hyperparameters for the dataset.
Sensitivity is the change in the E_{rms} value we are considering. This is like a point when there is no change in values and we can successfully determine that the currect area is a minima to given hyperparameters.
Initially starting the model at 0.1 LR we saw that the model was overshooting the minima, therefore the range at which is stops overshooting is defined by model 1. However, refering to the Fig5 we see that it has a steep descent, therefore we need to decrease the alpha further and we arrive at Model 2 where as we see in Fig7 it has converged properly. Increasing the sensitivity we will be able to get

Table 2: Linear Regression

Model	Alpha	Lambda	Sensitivity	E_{rms}
H-Obs Feature Concatenation	0.0003	0.5	$1x10^{11}$	0.5004
H-Obs Feature Concatenation	0.00003	0.5	0.001	0.5033
H-Obs Feature Substraction	0.00003	0.5	0.000001	0.5023
Oversampled H-Obs Feature Substraction	0.000003	0.5	0.000001	0.5016
GSC Feature Concatenation	0.000003	0.5	$1x10^{11}$	0.5026
Oversampled GSC Feature Concatenation with Epoch Shuffle	0.000003	0.5	$1x10^{11}$	0.5026
GSC Feature Substraction	0.000003	0.5	$1x10^{11}$	0.5020

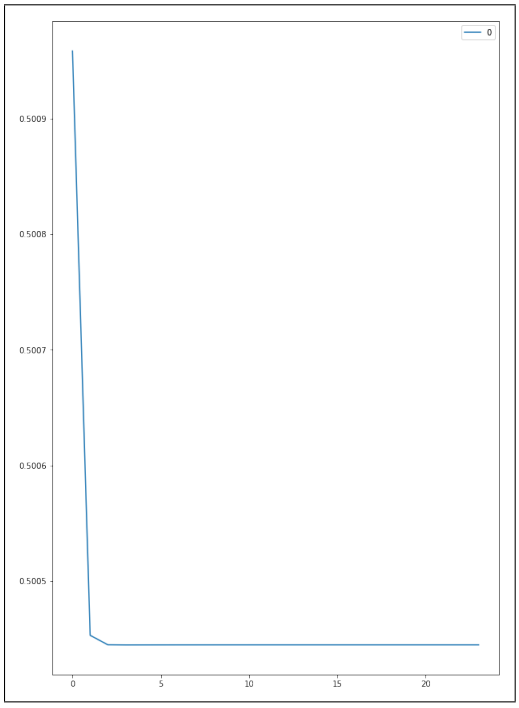

Figure 5: Validation E_{rms} for model 1 given in Table 2

same E_{rms} value as Model 1.
The results of the Experimentation[2] can be seen in Table[3] 2

3.2 Logistic Regression

We are going to add to more metrics while experimentation which are Precision and Recall. Basically, Precision is how useful the results are while recall is how complete the results are. Table 5 shows the final models which after changing hyper parameters we have obtained[4].

3.3 Neural Network

We Implement a Neural network on these four datasets and try to increase the accuracy. Since GSC has pixel values we hypothesize that GSC dataset will give us the maximum accuracy. We consider two models. One with dense layers and large amount of hidden units while the another being a small

[2]Models where Oversampling and Undersampling gave same results were ignored since they show that statistically there was no effect of repeating the samples

[3]H-Obs stands for Human Observed

[4]FC stands for Feature Concatenation, FS stands for Feature Substraction and * stands for values which are precomputed when it is taking too long to converge

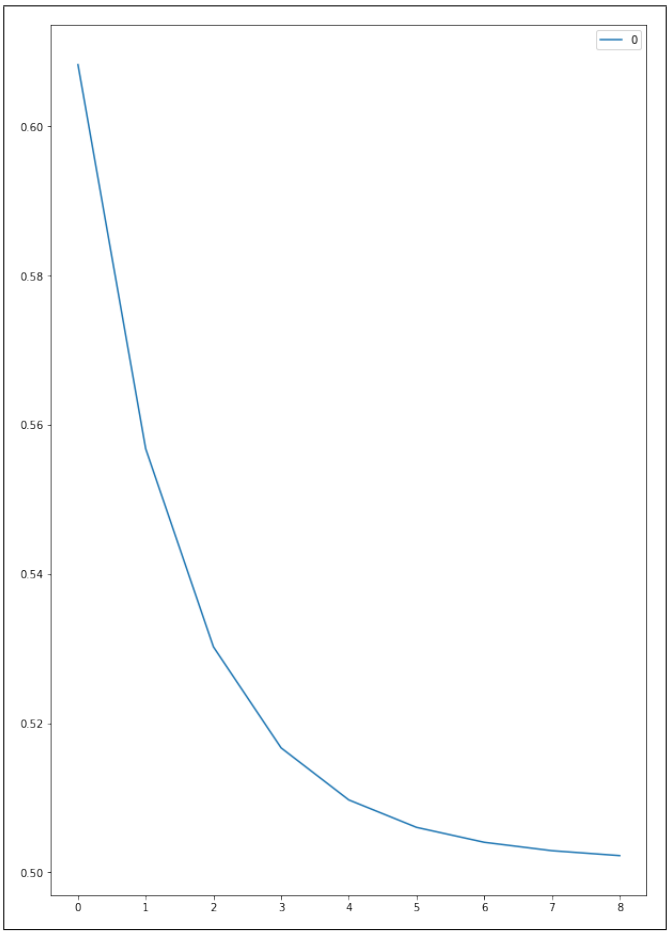

Figure 6: Validation E_{rms} for model 2 given in Table 2

Table 3: Logistic Regression

Model	Alpha	Lambda	Sensitivity	Accuracy	Precision	Recall
H-Obs FC	0.0000016	0.5	0.000000001	55.6962	0.5542	0.5822
H-Obs FS*	0.0000016	0.5	0.000000001	58.23	0.5867	0.5696
GSC FC*	0.0000003	0.5	0.000000001	61.4874	0.6176	0.6033
GSC FS*	0.0000003	0.5	0.000000001	72.5729	0.7476	0.6816

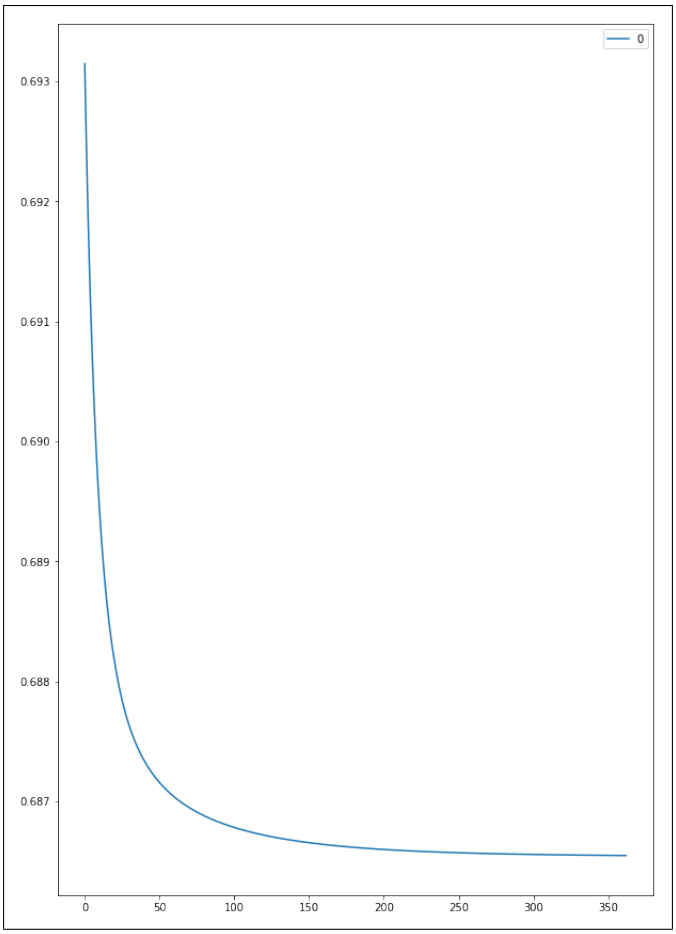

Figure 7: Cross Entropy Loss for model a given in Table 5

Table 4: Logistic Regression

Model	Dropout	Train Accuracy	Validation Accuracy
H-Obs FC with Two Dense Layers	0.3	0.83623	0.5139
H-Obs FS with Two Dense Layers	0.3	0.9234	0.5432
GSC FC with Three Dense Layers(1024 and 32)	0.3	0.93175	0.9156
GSC FC with Two Dense Layers(1024 and 32)	0.3	0.9874	0.9397
GSC FS with Three Dense Layers(1024 and 32)	0.3	0.92133	0.8491

neural network with High number of hidden units and small amount of layers.
We are using ‘Adam" optimizer since it gives better results.

4 Observation

4.1 Linear Regression

After changing hyperparameters we were unable to decrease the E_{rms} lower than 0.5. This could be because in a multidimensional plane, a linear function even after using Gaussian Model is unable to segment the data properly. If we take random data, sometimes there is high precision and Recall, however the accuracy is still at 50%, while the E_{rms} increases. This phenomenon because after rounding off the linear regression prediction we predict the same class but when we use its real values, it contributes more to the loss.
We have also observed that after overfitting the model it outputs a target value 2. Which doesn’t exist

Table 5: Best Models for based on Test Accuracy

Model	Test Accuracy
GSC FC with Three Dense Layers(1024 and 32)	93.38
GSC FC with Two Dense Layers(1024 and 32)	94.25

in the data. This is due to absence of a sigmoid function which rounds such values to 1.
The Epoch Shuffle method 2.4 doesn't give good results while training for GSC Dataset.

4.2 Logistic Regression

Based on the Observations obtained from training a Linear Gaussian Model, we perform better in Logistic Regression. As we see that the Accuracy has increased. The Human Observed Dataset accuracy isn't increased as significantly as the GSC dataset. The maximum accuracy which we can find after running the code for around 2 hours on AWS EC2 C5.4xLarge instance was 72.5729 for GSC Feature Substraction Dataset. This might even further decrease if kept for long time. The problem is therefore the time taken to train the model.

4.3 Neural Network

We see that the accuracy obtained for neural networks is the highest and Neural networks performs better in terms of pixel data. However, the Human Observed dataset overfits the model. But this overfitting is due to less amount of validatation and test data.
We see that Model with less dense layer tries to overfit the data. Therefore for GSC dataset setting the model with three dense layers works the best. However, the accuracy obtained on the Test dataset is more.

5 Conclusion

In our research we have found out that neural network model gives better accuracy and takes less amount of time to train compared to Linear and Logistic models. The fact that GSC dataset has pixel values makes it easier to train for Neural network. Human Observed dataset has low accuracy on all three methods, this might be due to ambigious features used to encode the featurs. Example: The label 4 signifies no specific pattern of handwriting. This is included both in positive and negative samples, this makes the the model hard to train.

References

[1] Elbow method (clustering) - Wikipedia. (2018) Retrieved November 01, 2018, from https://en.wikipedia.org/wiki/Elbow_method_(clustering)

[2] Jason Brownlee On. (2018) 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. Retrieved November 01, 2018, from https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

[3] Shirin Glander. (2018) Dealing with unbalanced data in machine learning. Retrieved November 01, 2018, from https://shiring.github.io/machine_learning/2017/04/02/unbalanced

[4] Stochastic gradient descent - Wikipedia. (2018) Retrieved November 01, 2018, from https://en.wikipedia.org/wiki/Stochastic_gradient_descent