ISYE 7406

Data Mining & Statistical Learning

Hassan Moughanieh

HW1

1. Introduction

1.1 Problem Description

Normalized handwritten digits are automatically scanned from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; the images have been deslanted and size normalized, resulting in 16 x 16 grayscale images (Le Cun et al., 1990). In this project we are trying to build a classification model that could identify the number as either 2 or 7 based on the 256-pixel information (16 x 16).

1.2 Dataset

The data are divided into two files, and each line consists of the digit id (0-9) followed by the 256 grayscale values between -1 representing black and +1 representing white. There are 7291 training observations and 2007 test observations.

2. Exploratory Data Analysis

In this project, we are only interested in datapoints related to 2's & 7's. Thus, the original datasets are filtered accordingly, resulting in a training dataset, ziptrain27, with 1376 datapoints, and a testing dataset, ziptest27, with 345 datapoints. Both datasets consist of 256 features pertaining to greyscale info and 1 additional feature, V1, representing digit id.

The data in ziptrain27 is split into 731 & 645, 2's and 7's, respectively (fig. 2.1).

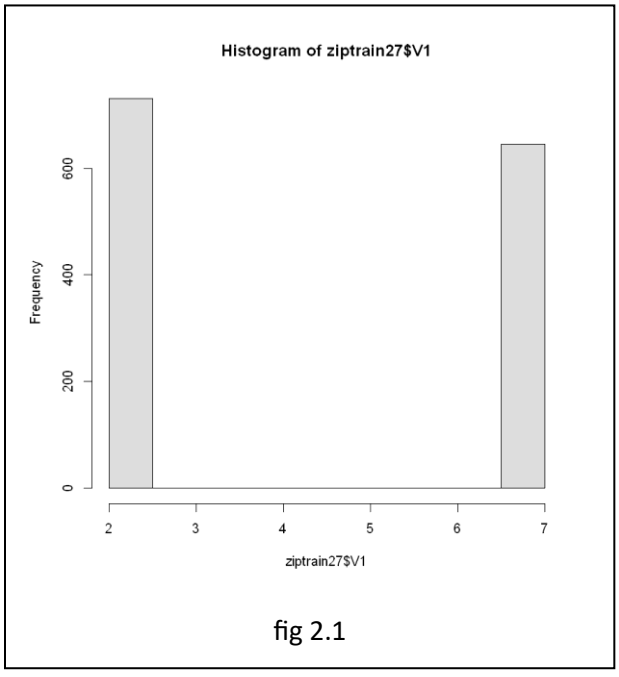

fig 2.1

We constructed an image using the greyscale values of datapoint #5, which happens to be a 7. The figure shown below confirms that finding (fig 2.2).

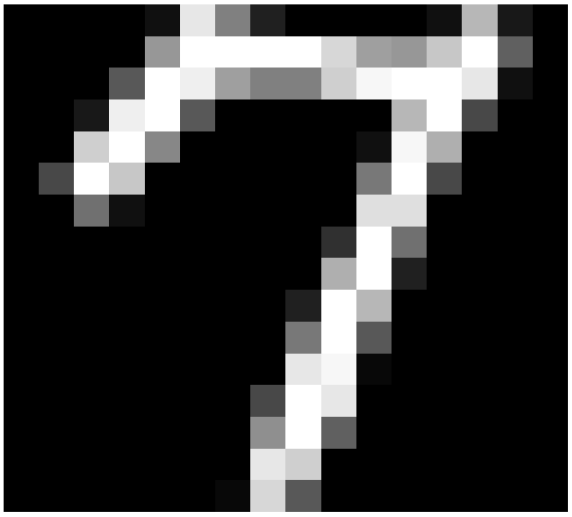

Fig 2.2

This figure complements the findings in the histogram (fig 2.3) of all the values from V2 to V257. Most of the values are around -1 (black) with fewer than 5000 values around 1 (white) and the remaining are distributed over that range.

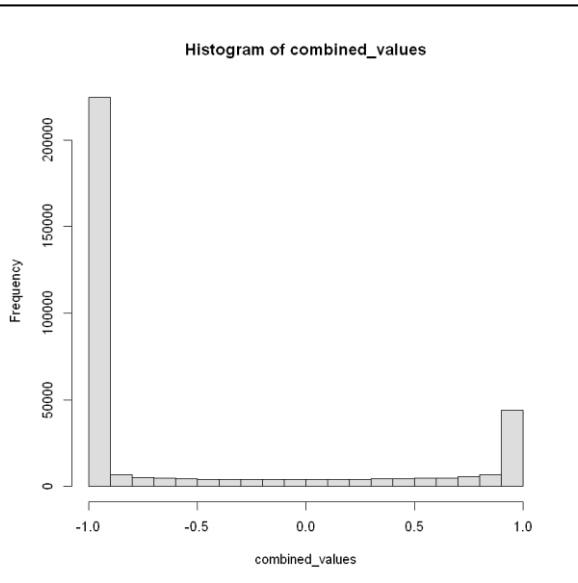

Fig 2.3

By looking at the summary of the dataset, we find that most of the features' values range between -1 & +1 as expected.

3. Methods/Methodology

The objective of this project is to build a predictive model that would determine if the scanned number is either 2 or 7 based on the 16x16 greyscale values. To achieve this, we will build linear regression & KNN models and compare their accuracy. We will also perform cross validation to make sure there are no biases in the models. The results of all the models will be compared at the end and the accurate model will be chosen.

A linear regression model was built using all features in the training dataset, ziptrain27, and then the model was used to predict the responses using the same dataset. 4.5, was used as a 50% threshold to determine if the results of prediction are 2's or 7's. After that, the predictions were compared against the real responses to calculate the training error.

After that, a K Nearest Neighbors model was built using the same dataset with k values of 1, 3, 5, 7, 9, 11, 13 and 15. The same training dataset was used for prediction and the outcome was then compared to the real responses to calculate the training error for each k value. The means training errors for the models came out as follows:

Model	Mean Training Error
Linear Regression	0.00072674
KNN1	0
KNN3	0.01017442
KNN5	0.01235465
KNN7	0.01453488
KNN9	0.01598837
KNN11	0.01598837
KNN13	0.01744186
KNN15	0.01744186

It is also clear from fig 3.1 that K =1 resulted in the lowest training error.

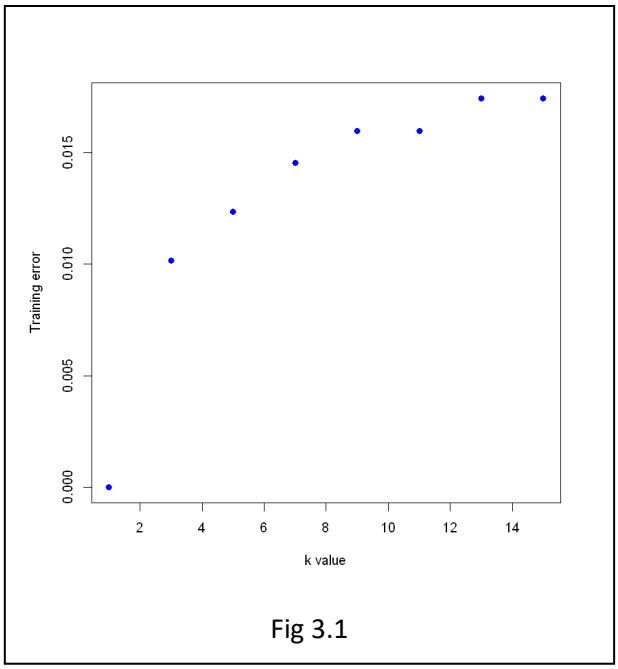

Fig 3.1

In order to accurately estimate the accuracy of the model, it should be tested using a testing dataset. A dataset that was not used in the development of the model. For this reason, the above models were then tested against a testing dataset, ziptest27. In this round, the testing error of the linear regression and KNN models were as follows:

Model	Mean Testing Error
Linear Regression	0.01739130
KNN1	0.01739130
KNN3	0.01449275
KNN5	0.01449275
KNN7	0.01739130
KNN9	0.01739130
KNN11	0.01739130
KNN13	0.02028986
KNN15	0.02028986

It can also be shown in fig 3.2 that k values of 3 or 5 produce the lowest testing error.

Due to the small size of the dataset (1376 observations), cross validation was needed to ensure the robustness of the model. In this project, we used Monte Carlo cross validation with 100 repetitions. Both the linear regression and the KNN models were cross validated. The training and testing datasets were

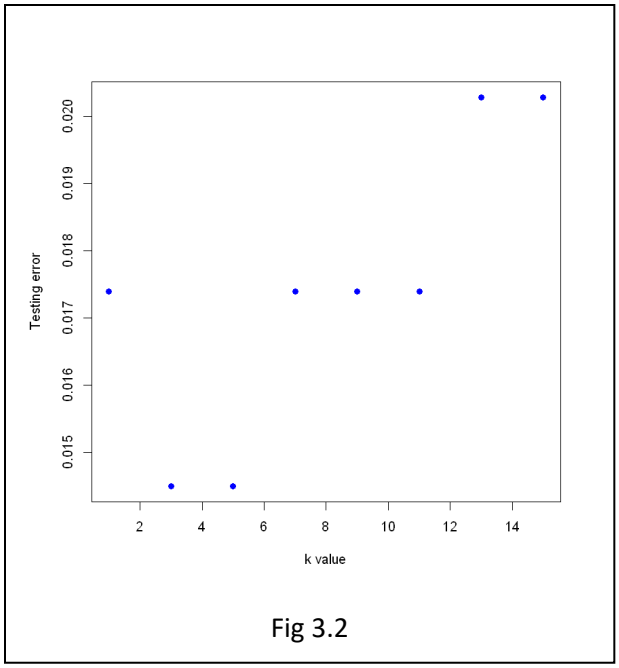

Fig 3.2

randomly sampled for each repetition from the full dataset (ziptrain27 + ziptest27) in the same proportions (1376 training & 345 testing datapoints).

With cross validation, the regression model had a mean testing error of 0.0114492753623188 while KNN had an increasing error with increased k values (fig 3.3).

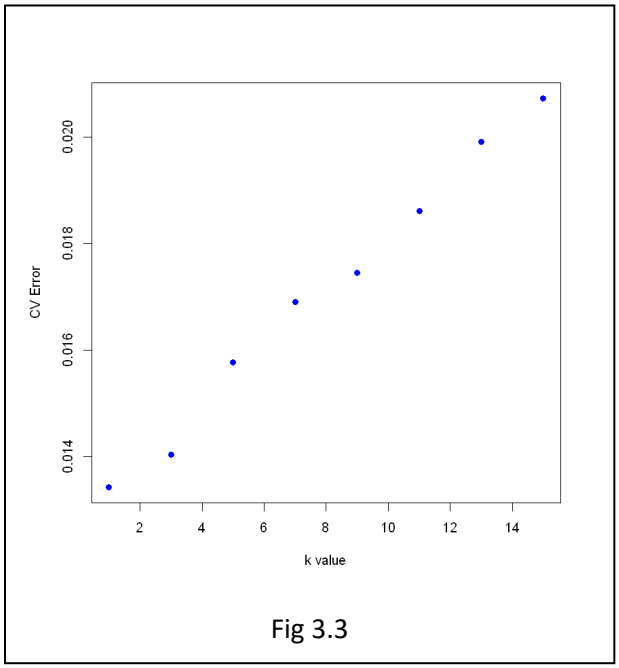

Fig 3.3

Figs 3.4 and 3.5 show how mean cv testing error and variance compared across all the models. The variance is a measure of spread. Larger variance indicated that the results had a larger range in the 100 repetitions of cross validation. Both figures show a superior performance of linear regression model.

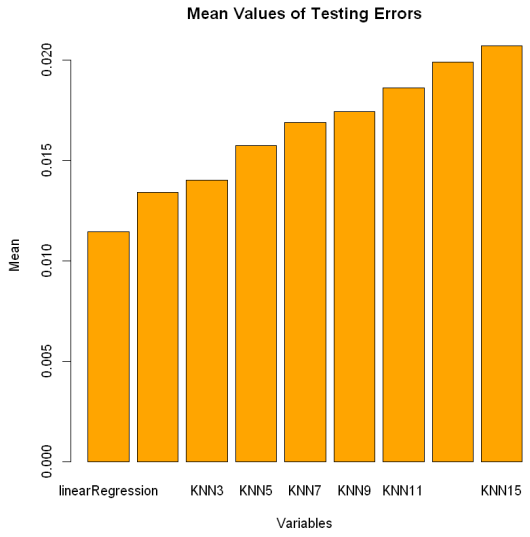

Fig 3.4

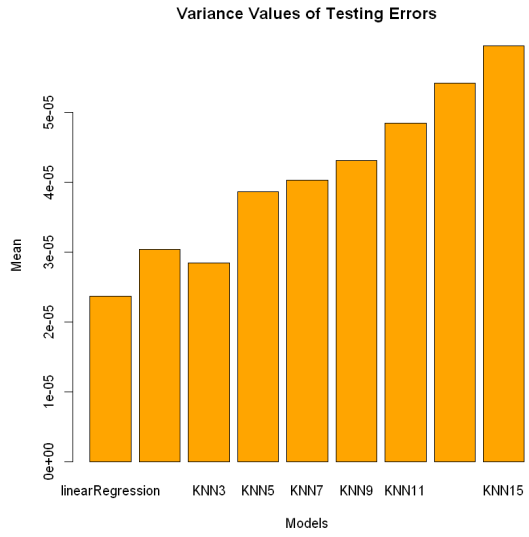

Fig 3.5

4. Findings/Conclusion

The analysis output is summarized in the table below:

	Mean Training Error	Mean Testing Error	Mean CV Testing Error
Linear Regression	0.00072674	0.01739130	**0.01144928**
KNN1	**0.00000000**	0.01739130	0.01342029
KNN3	0.01017442	**0.01449275**	0.01402899
KNN5	0.01235465	**0.01449275**	0.01576812
KNN7	0.01453488	0.01739130	0.01689855
KNN9	0.01598837	0.01739130	0.01744928
KNN11	0.01598837	0.01739130	0.01860870
KNN13	0.01744186	0.02028986	0.01991304
KNN15	0.01744186	0.02028986	0.02072464

From the summary table above, we see that KNN1 was the best performing model when measured by the training error. KNN model with k=1 is very sensitive to noise as it takes only 1 neighbor to make the classification. This can mean that the model is biased. Taking this into consideration, the result training error of 0.0 is not surprising as this is an indication that the model is overfit. Hence, training error is not a good measure for the performance of a model since it is tested on the same data that was trained on. When testing against the testing dataset, KNN3 & KNN5 equally outperformed the rest of the models with mean testing error of **0.01449275**. However, since the dataset is relatively small (1376 datapoints) cross validation was needed to ensure the robustness of the model and avoid bias. Surprisingly, this time linear regression model outperformed KNN in classification with a testing error of **0.01144928**.

Appendix

The analysis using R in attached for reference.