CP8318/CPS803 Machine Learning

Assignment 2

Rohaan Ahmed
PhD Student - Computer Science
rohaan.ahmed@ryerson.ca

November 5, 2020

Instructor: Dr. Nariman Farsad
November 5, 2020

1. Logistic regression and Gaussian Discriminant Analysis

1.1. Coding problem: Logistic Regression

Dataset 1

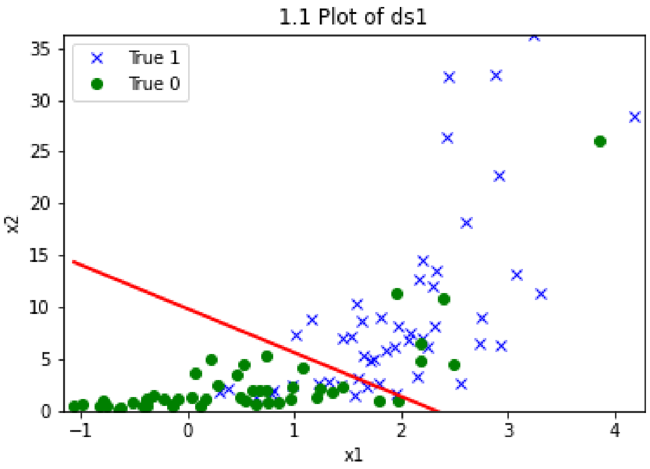


Figure 1:

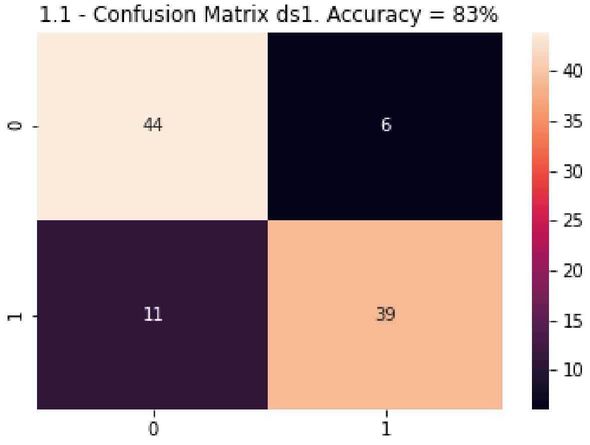


Figure 2:

Dataset 2

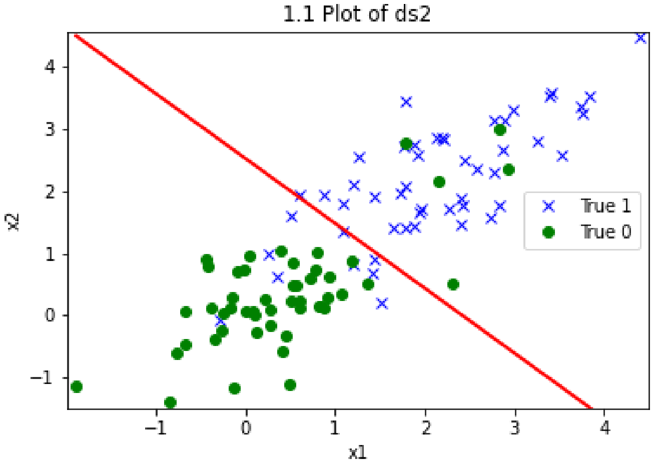


Figure 3:

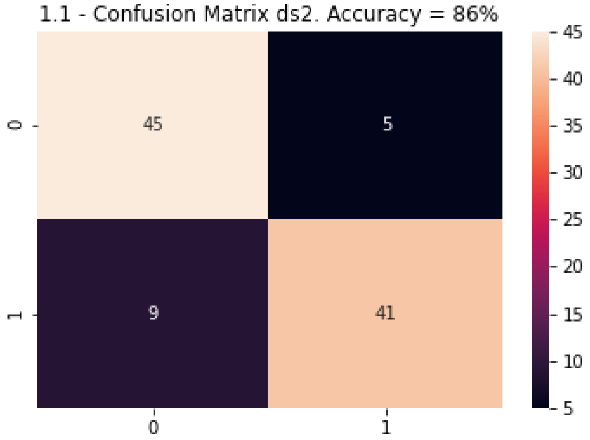


Figure 4:

1.2. Coding problem: Gaussian Discriminant Analysis

Dataset 1

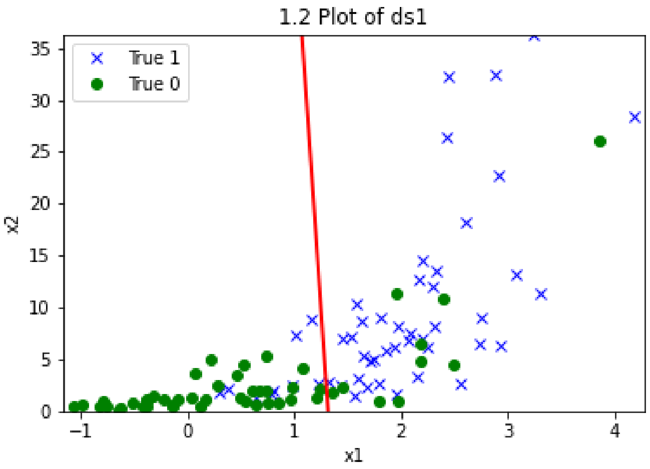


Figure 5:

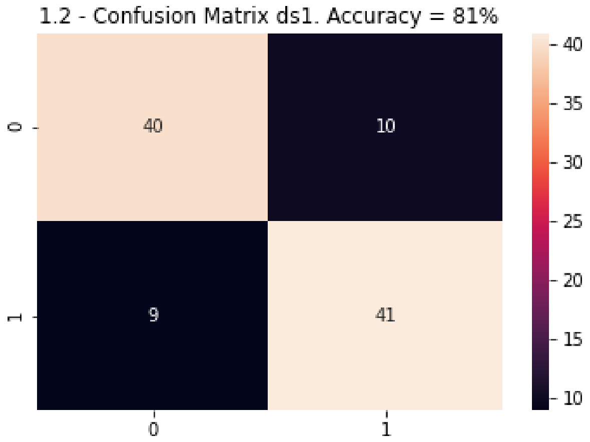


Figure 6:

Dataset 2

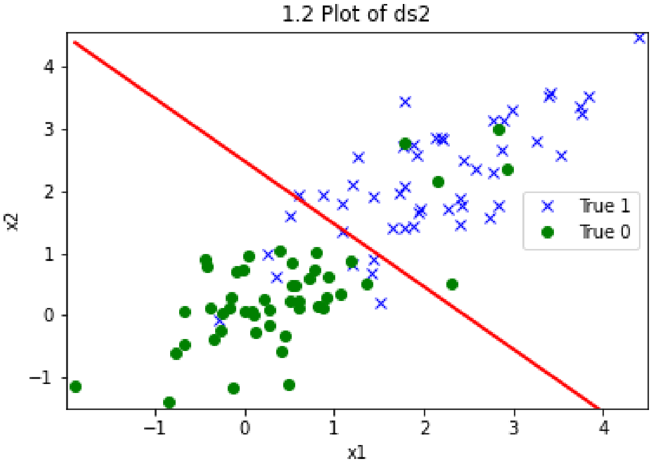


Figure 7:

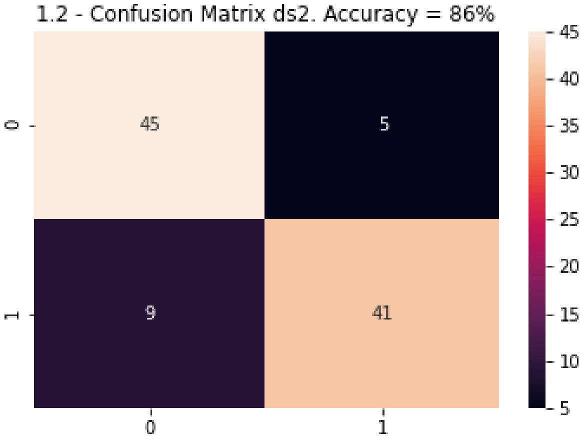


Figure 8:

1.3. Dataset 1 Comparison

For Dataset 1, Logistic Regression performs much better than GDA, as is represented by the Accuracy of the Confusion Matrix. The decision boundary obtained using Logistic Regression divides the datapoints using both x_1 and x_2, whereas GDA places far greater emphasis on x_1. This may be because GDA makes certain *strtict* assumptions about the data, including its Gaussian properties, which may not be sufficiently satisfied by the dataset to give a good enough model.

It can be seen from the plot that the variance of *True 0* datapoints and *True 1* datapoints is not similar, which further goes against the assumption of common covariance, i.e., the average squared distance from the mean for the datapoints is not similar, as is typically the case with Gaussian distributions.

1.4. Dataset 2 Comparison

For Dataset 2, the classifications obtained using Logistic Regression and GDA are very similar. It can be seen from the plot that the variance of *True 0* datapoints and *True 1* datapoints is much more similar than in Dataset 1, which more closely meets the GDA assumption of common covariance. This is represented by the Accuracy of the Confusion Matrix.

1.5. CP8318 Only Question

For Dataset 1, Logistic Regression performed significantly better than GDA. We can try a few techniques to improve our model.

a. Adjusting via Normalization

GDA assumes common covariance and Gaussian distribution for each feature variable. One way of ensuring this is to normalize the input features.

$$x \leftarrow \frac{x - \mu_x}{\sigma_x} \tag{0.1}$$

where:
x: is the feature matrix
μ_x is the mean of x
σ_x is the standard deviation of x

We test the above hypothesis on the two datasets. We follow the following steps:

1. Normalize the training dataset features
2. Fit the model to the training dataset and obtain the trained θ
3. Normalize the evaluation dataset features
4. Predict using the trained θ and normalized evaluation features

In general, GDA makes more specific assumptions about the datasets than Logistic Regression. When those assumptions are met, GDA can be expected to work better than Logistic Regression for classification. On the other hand, since Logistic Regression makes fewer assumptions about the dataset, it is more generally applicable on real world datasets, such as when the probability distribution of the features is not Gaussian.

The results, shown in Figures 9 and 10, show that normalizing the input features improves the predictive power of the GDA, but not significantly. In other words, when the common covariance assumption of the GDA is true, it does a slightly better job of learning and predicting. However, despite this, Logistic Regression still performs better on the given datasets.

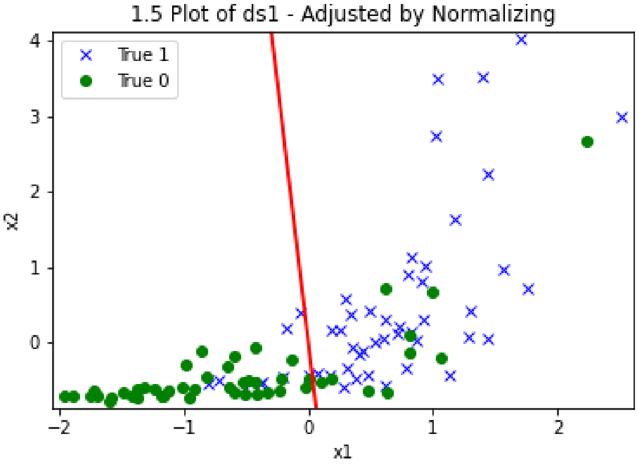


Figure 9:

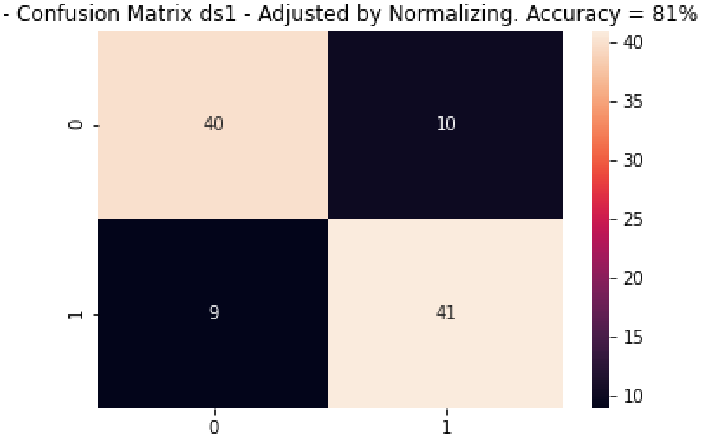


Figure 10:

b. Adjusting by Ignoring Outliers

Next, we delete the "outliers" in the dataset and train the model only on data which meets the condition below:

$$x_k \leftarrow |x_k^{(i)} - \mu_{x_k}| < m\sigma_{x_k} \tag{0.2}$$

where:
x_k: is the feature vector

μ_{x_k} is the mean of x_k
σ_{x_k} is the standard deviation of x_k
$m \in \mathbb{R}^+$, (0 or positive real number), a scaling factor

That is, we consider the points beyond a certain radius, $m\sigma_{x_k}$, around the mean of the feature vector, μ_{x_k}, to be "outliers". This allows us to train on data that is more *representative* of the *usual* condition, and ignore the datapoints which are potentially *anomalous*

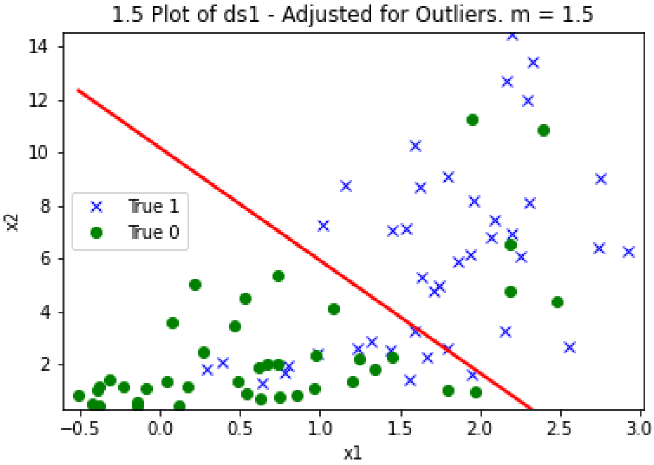


Figure 11:

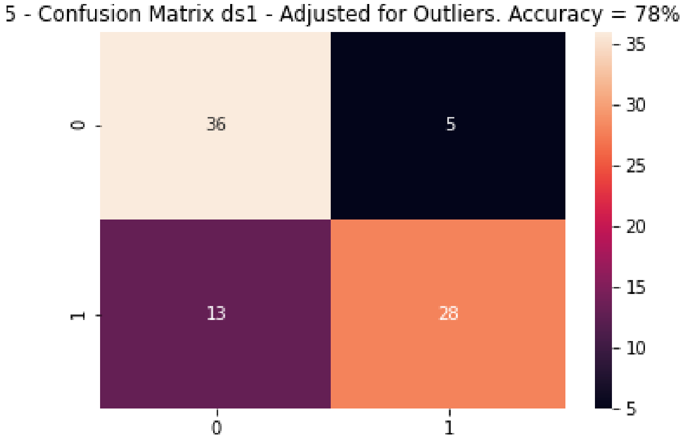


Figure 12:

The results, shown in Figures 11 and 12, show that by ignoring outliers, we achieve a better fitting decision boundary.

c. Adjusting via Feature Mapping

Taking a look at the datapoints distribution, we can see that it roughly resembles the shape of an exponential function. We can use this to our advantage by using feature mapping to map the data to a logarithmic function, which may provide us with a more linearly separable decision boundary.

We will map the function as follows:

$$\hat{x} = ln(x^2)$$

where:
$\hat{x}$ is the set of new features obtained after mapping
x is the original set of features

The mapping above was obtained after some trial and error. The resulting model can be seen in Figure 13 and 14. As can be seen, both the separation of datapoints as well as the accuracy improve.

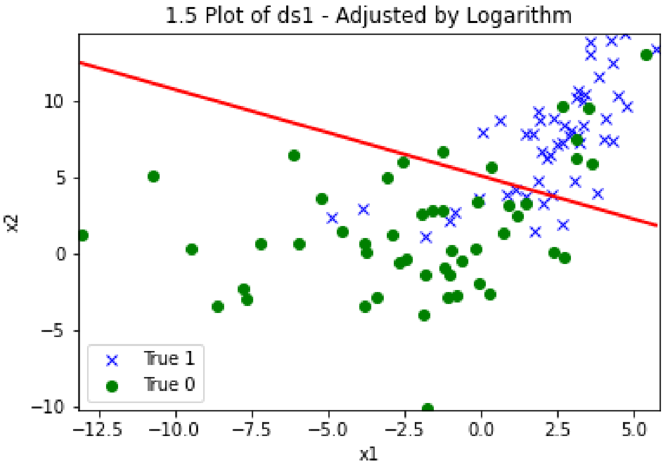


Figure 13:

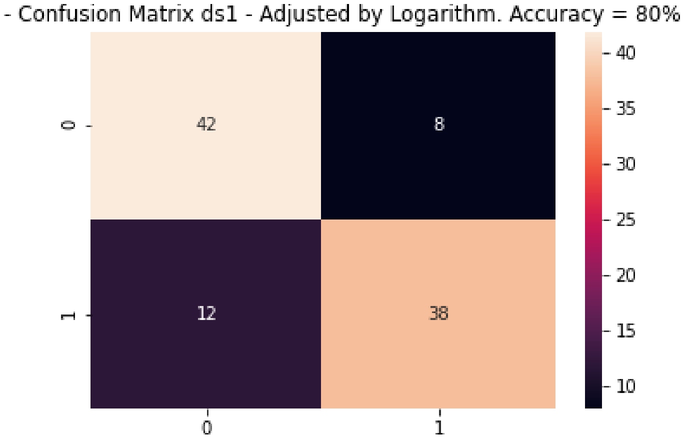


Figure 14:

d. Other techniques

Other transformations which may be applied to the data in order to obtain a better decision boundary are:

- Other types of Feature Mapping techniques
- Weight-ing the Datapoints: This would be done by multiplying datapoints with a weight depending on how close they are to the mean of the vector. This is similar to what we did above in the Ignoring Outliers technique.
- A combination of the techniques above. For example, we can reject outliers, normalize the data, and then feature map it.

We leave the testing of these techniques as future work

2. Poisson Regression

2.1. CP8318 Only Question

The Poisson distribution can be written in the form of an exponential family function using $a = e^{\log a}$

$$\begin{aligned} p(y;\lambda) &= \frac{\exp(-\lambda)\lambda^y}{y!} && (0.3)\\ &= \exp\log(\frac{\exp(-\lambda)\lambda^y}{y!}) && (0.4)\\ &= \exp\left(-\lambda + y\log(\lambda) - \log(y!)\right) && (0.5)\\ &= \frac{1}{y!}\exp(\log(\lambda)y - \lambda) && (0.6) \end{aligned}$$

where:

$$\begin{aligned} b(y) &:= \frac{1}{y!} && (0.7)\\ \eta(\lambda) &:= \log(\lambda) && (0.8)\\ T(y) &:= y && (0.9)\\ a(\eta) &:= \exp(\eta) = \lambda && (0.10) \end{aligned}$$

2.2. CP8318 Only Question

The canonical response function maps η to the expected value $\mathbf{E}[T(y);\eta]$, thus:

$$\mathbf{E}[y;\eta] = \lambda \tag{0.11}$$

Based on the fact that $\eta(\lambda) = \log(\lambda)$, the exponential function maps $\eta(\lambda)$ to $\mathbf{E}[y;\eta]$. Thus, the exponential function is the canonical response function:

$$g(\eta) = \mathbf{E}[y;\eta] = e^{\eta} = e^{\theta^T x} \tag{0.12}$$

2.3. Coding problem

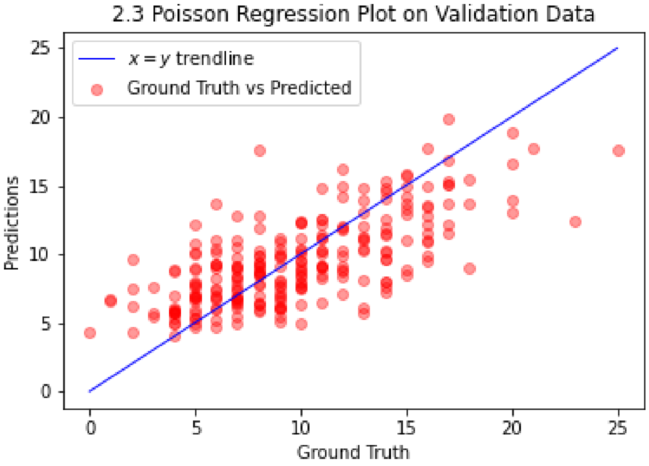
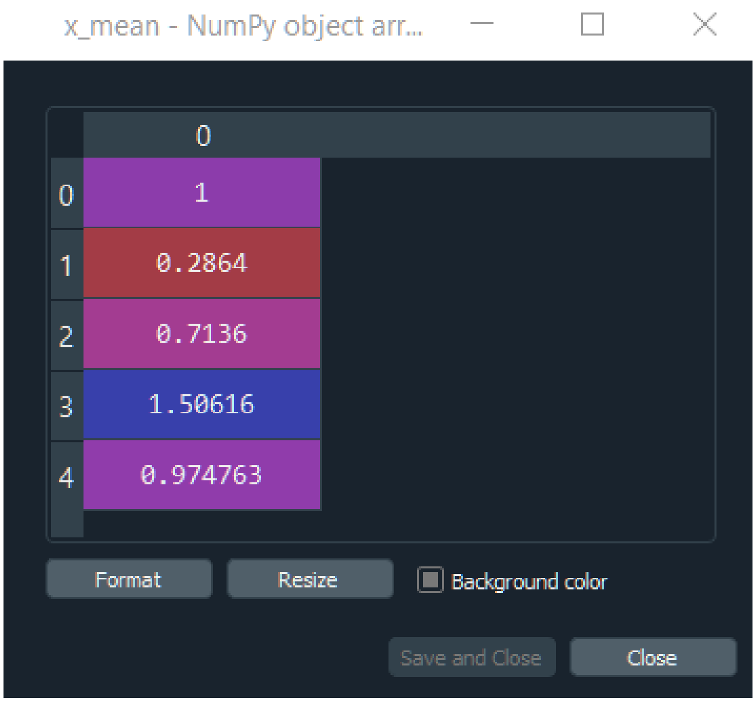


Figure 15:

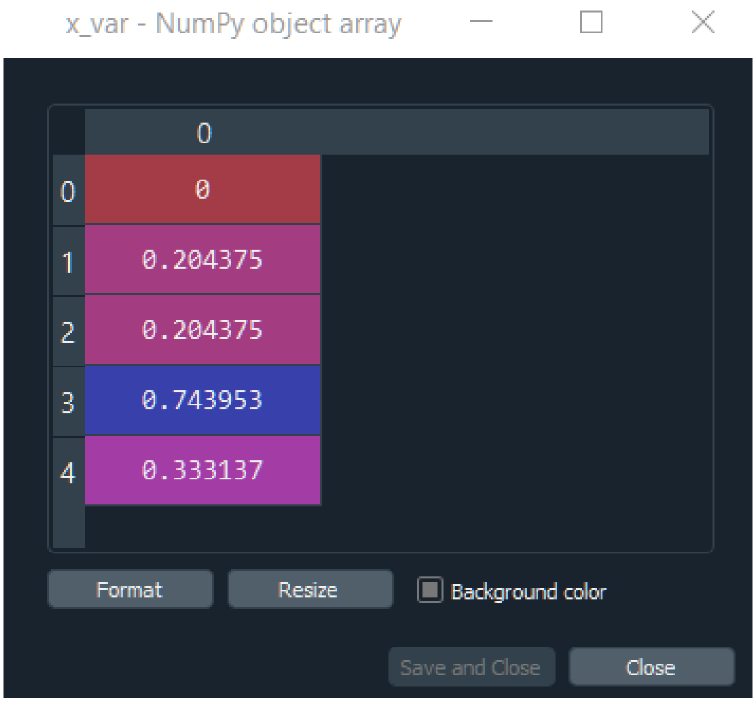
Figure 15 shows the predicted values vs the ground truth in red, along with the $x = y$ trendline in blue. The trendline depicts the ideal case, where the predicted and ground truth values are always equivalent. The predicted values follow the trendline approximately, but do not match it perfectly. Poisson Regression differs from Linear Regression in the following ways:

1. Assumes that the errors follow a Poisson distribution, rather than Gaussian.
2. Instead of modeling y as a linear function of the regression coefficients θ, we model the natural logarithm of the response variable, $ln(y)$, as a linear function of the coefficients, θ.
3. Assumes that the mean and variance of the errors are equal



	0
0	1
1	0.2864
2	0.7136
3	1.50616
4	0.974763

Figure 16: Training Set Feature Mean Values



	0
0	0
1	0.204375
2	0.204375
3	0.743953
4	0.333137

Figure 17: Training Set Feature Variance Values

Figure 16 shows the Mean of each of the x variables in the training dataset, and Figure 17 show the corresponding Variance. As can be seen, assumption (3) above is not satisfied with the given training dataset. Thus, it is not a perfect Poisson distribution, and cannot be expected to give great prediction results using a Poisson Regression Model.

3. Spam Classification

The results obtained by completing and running the code for this section are printed below:

```
Size of dictionary: 1722
Naive Bayes had an accuracy of 0.978494623655914 on the testing set
The top 5 indicative words for Naive Bayes are: [‘claim’, ‘won’, ‘prize
    ’, ‘tone’, ‘urgent!’]
The optimal SVM radius was 0.1
The SVM model had an accuracy of 0.9695340501792115 on the testing set
```

3. Spam Classification Revisited with Punctuation Normalization

We revisited Section 3 to answer the question: what happens when we normalize the messages by deleting punctuations from them, in addition to converting them to lower-case. We hypothesize that this may result in a better classifier, since punctuations can dilute the associations obtained with certain words.

The results of the experiment are printed below:

```
EXPERIMENT: WHAT HAPPENS WHEN WE DELETE PUNCTUATION FROM OUR MESSAGES?
Size of dictionary: 1552
Naive Bayes had an accuracy of 0.9838709677419355 on the testing set
The top 5 indicative words for Naive Bayes are: [‘claim’, ‘prize’, ‘won
    ’, ‘tone’, ‘guaranteed’]
The optimal SVM radius was 0.1
The SVM model had an accuracy of 0.967741935483871 on the testing set
```

The results can be summarized as follows:

- The size of the dictionary reduced slightly, from 1722 to 1552, or app. 10%. This would reduce both compute time and storage requirements by a small, but not insignificant, factor.
- The accuracy of the Naive Bayes classifier improved slightly (app. 1%).
- The optimal radius for SVM, and the best case accuracy, remained relatively unchanged

- The list of Top 5 words indicating a Spam email changed, both in terms of content and strength.
  - The word “urgent!” was replaced by the word “guaranteed”
  - The words “prize” and “won” switched orders

Thus, it can be said that normalizing based on punctuation provides a marginally better classifier.

Based on these results, we hypothesize that if we were to base our classification on the “root” words contained in the messages, we may obtain an even better classifier. For example, considering the words “urgent” and “urgently” as one may improve our spam classification. Such normalization techniques are often used in industry, for example “stemming” and “lemmatization”, and are here left for future work.