Statistical Learning of Joint versus Conditional Probability Distributions

Logistic Regression Compared to Naive Bayes for Classification Employing Categorical Features

Efklidis Katsaros, s2023318

I. Introduction

In terms of this study, we discuss on Generative versus Discriminative classifiers. The former are defined as the ones modelling the joint probability $p(x, y)$, where x are the inputs and y are the labels. Once the joint probability distribution is learnt, predictions are generated employing the Bayes formula to compute the conditional probabilities $p(y|x)$ for each class. On the contrary, Discriminative classifiers model the posterior probability $p(y|x)$ for each class directly, that is, they learn a function of the data yielding probabilities for a new instance belonging to a specific class. Discriminative classifiers' prevalence is reflected on Vapnik's statement, according to which, "one should solve a problem directly and never solve a more general problem as an intermediate step".

In this note, we experiment with the Logistic Regression (LR) and Naive Bayes (NB) classifiers as a characteristic and comprehensible example of a Discriminative- Generative pair. We provide the theoretical infrastructures upon which the two models learn the training data, build their individual rules as a function of the inputs and yield predictions. Furthermore, we benchmark both learning methods. Three Naive Bayes classifiers are trained employing different smoothing parameters. Additionally, three Logistic Regression classifiers are formulated, namely, one employing the full features' set, one using a subset of the features and a Ridge Logistic Regression on the full feature space. That being said, we monitor the error rate for each classifier, in terms of the 0/1-Loss, asymptotically for an increasing sample size. Consequently, we discuss on and compare the results linking them with the classifiers' theoretical foundations.

II. Data and Evaluation

The dataset upon which comparisons are conducted is named after "Car Evaluation Database"[1]. It consists of 1728 instances and 6 categorical features. The four initial classes (unacc, acc, good, v.good) are merged into two, namely Negative and Positive. Then, the dataset is split into two subsets, the train and the test set, each comprising of 864 instances.

More specifically, half of the features take on four different labels (levels) and the other half of them are assessed on a three-level scale. Note that, features are essentially ordinal (see Table 1). Still, for Logistic Regression, we do not assume any linear relationship between the linear predictor and the features; Instead, we estimate different parameters for each level obtaining a model with

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.names

Feature	Description	Levels
class	car acceptability	{Negative, Positive}
buying	buying price	{v-high, high, med, low}
maint	price of the maintenance	{v-high, high, med, low}
doors	number of doors	{2, 3, 4, 5-more}
persons	capacity in terms of persons to carry	{2, 4, more}
lug_boot	the size of luggage boot	{small, med, big}
safety	estimated safety of the car	{low, med, high}

Table 1: Caption

more expressive power that could potentially be more prone to over-fitting on smaller sample sizes.

The Generative-Discriminative pair of classifiers is assessed through the 0/1 loss (misclassification rate, error rate), which is the complement of accuracy, or,

$$error(n) = \frac{\sum_{i=1}^{n} I(\hat{y}_i \neq y_i)}{n},$$

where I is the indicator function, $\hat{y}_i$ are the predictions, y_i are the true labels and n is the sample size. Error is monitored on the full test set. Classifiers are trained iteratively on train sets of ascending size. That said, for each classifier and for each sample size (n = 1, ..., 864), we obtain one 0/1 loss value. Losses are averaged over 20 iterations, so that their estimates are corrected for the random splits of the dataset. Finally, 6 curves of the error are plotted across the sample sizes.

III. Naive Bayes Classifier

The Naive Bayes classifier is a Generative one, assuming that given the class, features are independent. That is, the covariance matrix is diagonal. While the assumption is generally not met, derived simplifications facilitate density estimations. Naive Bayes outperforms many refined alternatives on specific tasks, as the densities-introduced bias is not affecting the predictions.

The Bayes rule is used to predict the class y implicitly from the likelihood $P(x|y)$ and the prior $P(y)$:

$$\hat{y} = \operatorname*{argmax}_y P(y|x) = \operatorname*{argmax}_y P(x|y)P(y)$$

In order to reassure that naive Bayes yields probabilities for each class, a Laplace smoothing correction is incorporated in the formula. When the smoothing parameter α is set to be one, one fake instance is assumed for each class. That is, the naive Bayes will never yield zero probability estimates, even if there is no information about a feature value within a class. More formally, this is expressed as follows:

$$P(X_j = k|y) = \frac{n_{k,y} + \alpha}{n_y + \alpha K_j}, \tag{1}$$

where y denotes the class and K_j is the number of values that feature X_j can take. Moreover, n_y is the number of instances belonging to class y and $n_{y,k}$ represents the number of training instances where class is y the feature has value k. Note that for α small enough, parameters converge to their maximum likelihood estimates. Asymptotically, when α gets larger, probability converges to uniform density $\frac{1}{K_j}$. Different experiments were conducted to assess performance of such models. Smoothing parameter varied across the values of a set $\alpha \in \{1, 0.1, 10\}$.

IV. Logistic Regression Classifier

Logistic Regression is a Discriminative classifier, computing directly $p(y|x)$ as:

$$p(y|x) = \frac{1}{1 + e^{-X^T \beta y}}, \qquad y \in \{-1, 1\} \textit{ indicating the class}$$

Essentially, Logistic Regression is a linear classifier mapping the data points from the d-dimensional space to a single-valued linear predictor. The latter is wrapped up with the link (logistic) function to yield probabilities of belonging at each class. Classification is achieved though opting for the most likely class conditional to the features, or more formally,

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(y|x)$$

As an extension of linear regression, multicollinearity issues need to be accounted for unbiased estimation of the weights' standard errors. In our case however, the feature space is not continuous and thus linearly related predictors that would inflate the Logistic Regression standard errors do not need to be dealt with. On this discrete feature space the linear predictors are formulated through differences of means (effects' model).

A full and a reduced logistic regression model were fit on the training data, in the fashion described on the previous section. The full model fits a hyper-plane on the 6d space while the reduced one fits a line to bisect the area. Furthermore, we fit a third model, namely, Ridge Logistic Regression. That is, a L2-norm regularization term was introduced upon loss minimization. For each sample size, we perform 5-folds cross validation to pick the optimal λ out of a dense sequence ranging from 0 to 5. Subsequently, we fit both the regularized model and the unpenalized ones to evaluate performance.

Note that, for n small enough, weights β cannot be uniquely identified as the linear equations system has indefinite solutions as a result of a singular matrix. Once n gets larger, computations are feasible. Additionally, the model cannot "see" some levels of the categorical features during the training, that is, corresponding weights are not estimated. Subsequently, prediction on such examples is not plausible. Instead of coming up with an ad-hoc fix for each problem, we decided to skip cases for which one of the aforementioned caveats would occur. For this, we made use of the `tryCatch` function in **R**. Naturally, comparisons are sensible once the standard error of the 0-1 loss estimate (as obtained through the 20 iterations) gets small enough (roughly, after n=30).

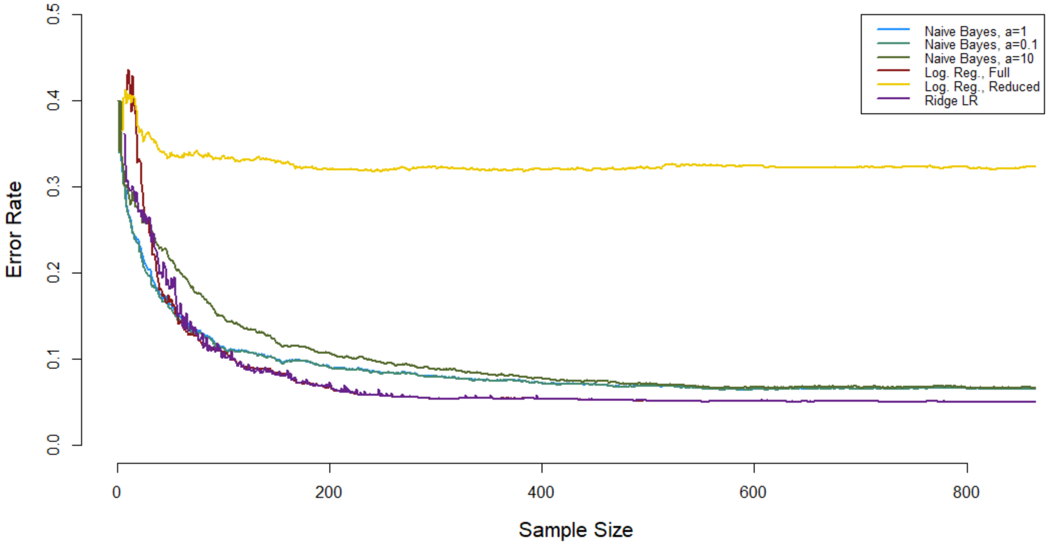

Figure 1: 0-1 Loss plotted across sequential ascending sample sizes.

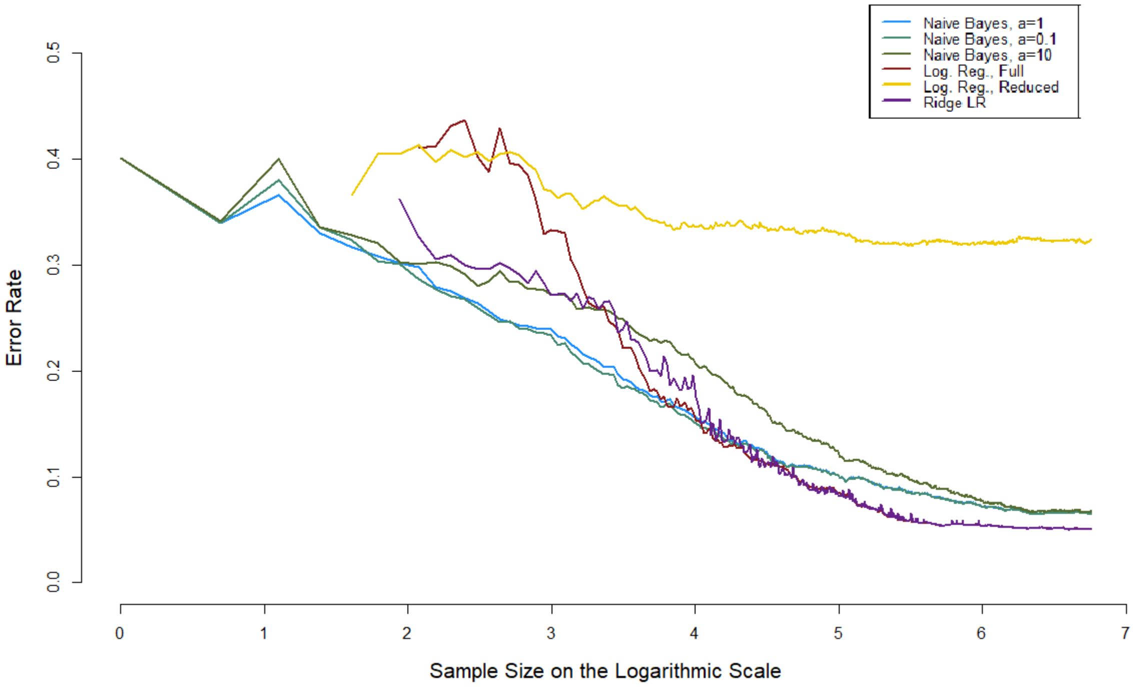

Figure 2: 0-1 Loss plotted across the logarithmic transformation of an ascending sample size.

V. Results

I. Comparisons

Error rates across different sample sizes are represented in Figure 1. To facilitate interpretation for smaller sample sizes, we also illustrate a logarithmic transformation of its x-axes, depicted in Figure 2. 0/1 Losses on the test set for n=864 are provided in Table 2. Clearly, LR and Ridge LR are the best performing classifiers with the latter yielding a slightly -but not significantly- lower error rate.

Classifier	**Error rate**
NB, $\alpha = 1$	0.0657
NB, $\alpha = 0.1$	0.0655
NB, $\alpha = 10$	0.0670
LR, 6-features	0.0504
LR, 2-features	0.3233
LR, Ridge	**0.0503**

Table 2: Error rates for n = 864. Ridge Logistic Regression outperforms naive Bayes and its unpenalized relatives.

Ridge LR was employed with a λ chosen over a grid including zero, that is, no penalty at all. Subsequently we would expect that its performance should be at least equivalent to the simple LR in the training set, which was, indeed, the case. In the test set, however, for specific values of n (3.5 - 4 on the log scale), we observe slight deviations due to random splits. We pick the optimal λ based on cross validating our training data but we are under-fitting the test set. This could be partly confronted by picking the optimal λ based on the "one-standard-error" rule as described by the authors of "Elements of Statistical Learning".

In contrast to the 6-features LR and the Ridge LR, the 2-features LR classifier produces higher error on the test set. When the sample size is small (n<20), reduced LR performs better than the full LR. The latter cannot learn the prediction rule as a function of the data as the weights are not stable yet. As soon as n get large enough (3 on the log scale), reduced LR is lead to asymptotic convergence whereas full LR keeps adjusting its decision areas. Clearly, data points are more separable on the 6d than the 2d feature space.

NB classifiers demonstrate equivalent performances asymptotically - highest performance, however, is noted for $\alpha = 0.1$. For small sample sizes, smoothing parameters $\alpha = 1$ and $\alpha = 0.1$ yield slightly lower errors than $\alpha = 10$ does. As explained in section 3, larger values of α affect predictive probabilities. When n gets larger (n>500), error rates tend to overlap. That is, for larger sample sizes, NB estimates converge to the Maximum Likelihood estimates.

Generally, NB classifiers tend to perform better than LR ones for smaller sample sizes (n<50). Estimating the parameters that maximize the likelihood of the joint probability is formulated as follows:

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} -logP_{\beta}(y_i, x_i) = \underset{\beta}{\text{argmin}}(\sum_{i=1}^{n} -logP_{\beta}(y_i|x_i) + \sum_{i=1}^{n} -logP_{\beta}(x_i))$$

Note that β is employed conventionally and denotes the NB parameters. The second term of the expression acts as regularizer and decreases variance. That being said, superior NB performance for smaller sample sizes is justified. Ridge LR is the sole discriminative classifier approaching NB in smaller sample sizes, as it also penalizes the loss function. Error rate is close to NB from the very beginning (see Figure 2, 2.3 on the log scale)

When n gets large enough, however, LR improves performance significantly, reaching lower error rates. This is in accordance with Andrew Y. Ng. et al. who show that, while NB classifiers yield lower 0/1 losses on smaller samples, LR tends to yield superior performance upon convergence. That is, LR reaches lower asymptotic error rates. Note that, for the specific dataset, convergence is achieved upon $n = 500$ for both methods (See Figure 1)

More formally, Andrew Y. Ng. et al. make use of the big-O notation to define express complexity of convergence. According to them, NB approaches its lowest error rate after $O(logn)$ rather than $0(n)$ number of training instances. In this context, convergence refers to estimating parameters that would only slightly deviate from their asymptotic values ($n \rightarrow \infty$).

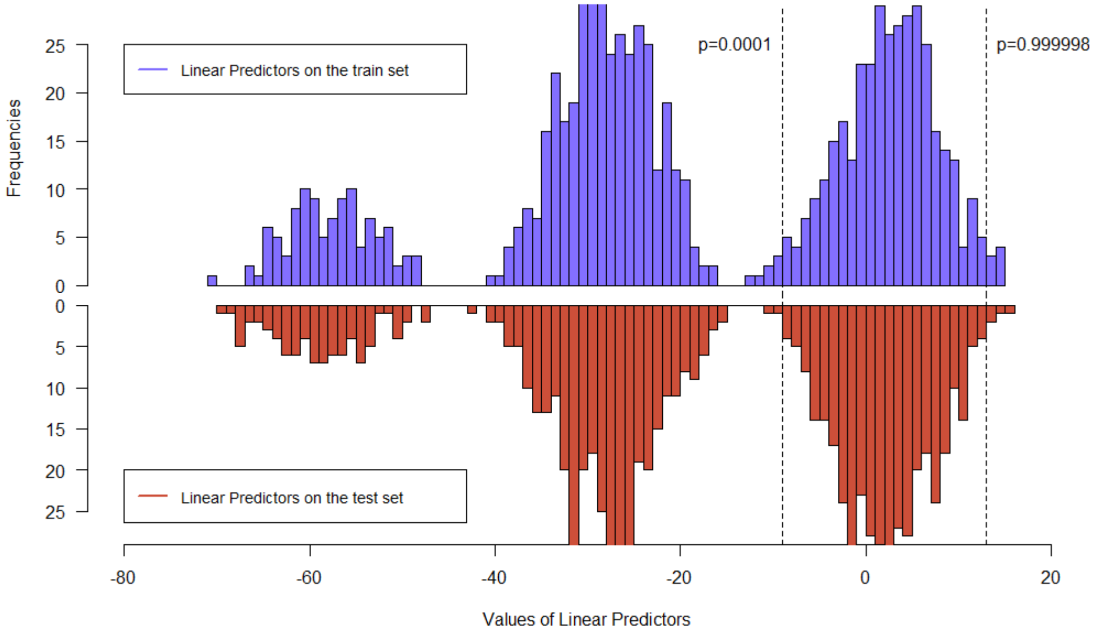

Figure 3: Linear predictors' histograms on the train and test set. Dashed lines stand for the values of linear predictors upon which, probabilities of classifying an instance as Positive are $p \leq 0.0001$ and $p \geq 0.999998$.

II. Interpretation

Accuracy achieved with Ridge LR and LR was 0.9497 and 0.9496 respectively, that is, we can categorically assess a car's acceptability in two classes with such probabilities of being correct. Difference between the classifiers is negligible for larger sample sizes. Still, for smaller ones, Ridge performs generally -but not always- better, because optimal λ underfits the test data for some sample sizes. In such cases however, NB is the winner, minimizing 0-1 Loss faster. Furthermore, should log-odds' extraction and coefficients' interpretation be of importance, LR would be the optimal classifier.

A linear decision boundary is satisfactory separating the classes. Still, adequacy of accuracies reported is closely related to the problem investigated. Should a percentage of 5% of the cars examined be considered too high in terms of costs, more intricate, non-linear classifiers would have to be considered.

III. Linear predictors

Monitoring the training and testing procedures, we observed that fitted probabilities are very close to 0 and 1. Accuracy on the train set however is not 1, that is the classes are not linearly separable. Quantifications of LR are expressed on the linear predictors as described in Section 4. Then, the logistic function yields probabilities. Reversely engineering our model's predictive probabilities on the train and the test set we can extract the linear predictors on the two sets respectively.

In Figure 3, one can observe the histograms of the linear predictors for both the sets. The dashed lines stand for the values yielding probabilities 0.0001 and 0.999998. Bearing in mind that logistic is a monotone function, we can easily infer that all values on the left hand side of the left dashed line, yield probabilities very close to zero and vice-versa. Subsequently, dubious instances are considered as the ones whose linear predictors are accumulated on the right-most histogram. For the rest, the logistic regression classifier is very "confident". Switching back to the initial 4 labels (ordinal; unacc, acc, good, vgood), we observe that LR is "confident" for the Negative class (unacc) whose linear predictors are detected in the two left-most histograms. The rest of the linear predictors are mapped to instances deriving from the Positive class (acc, good, vgood). Consequently, misclassifications occur mainly due to Positive labeled instances assigned to the Negative Class.

IV. Predictive probabilities

Predictive probabilities were monitored across the ascending sample size, so as to further examine the classifiers' behaviour on the test set. That being said, we opted for the best performing NB ($\alpha = 0.1$) and the 6-features LR classifier. For each of them and for each sample size, we examined:

- The mean predictive probability of each class , given that instances were correctly classified (see Figure 4).

  Measuring probabilities of correctly classified instances, is an indication of confidence for a classifier when making the correct decision. As that, we examined these probabilities for the increasing sample size. Both classifiers are confident to assign an instance to the Negative class (0) when the prediction overlaps with the ground truth. Assessed probabilities are above 0.95. LR is similarly confident for the Positive class (1). On the contrary, NB is more reluctant when assigning an instance to the Positive Class. Consequently, probabilities for NB start to differentiate soon enough, whereas for LR we need 200 training examples before some doubt for the Negative class occurs.

- The mean predictive probability for each label, no matter how they were classified (see Figure 5).

  Monitoring predictive probabilities of instances based -directly- on their label, we are able to observe the classifiers' certainty regardless of whether the prediction was accurate. Note that, the lower bound on the y-axes is no more 0.5, as erroneous predictive probabilities are allowed in this frame. LR is -again- confident to predict both classes. NB, on the other hand, tends to be uncertain for the Positive labels; This is mainly depicted on smaller sample sizes. NB is overwhelmed by its prior probabilities in this context. Based on these plots, one can say that LR balances its misclassifications between the two classes whereas NB is prone to assign a Negative instance to the Positive class.

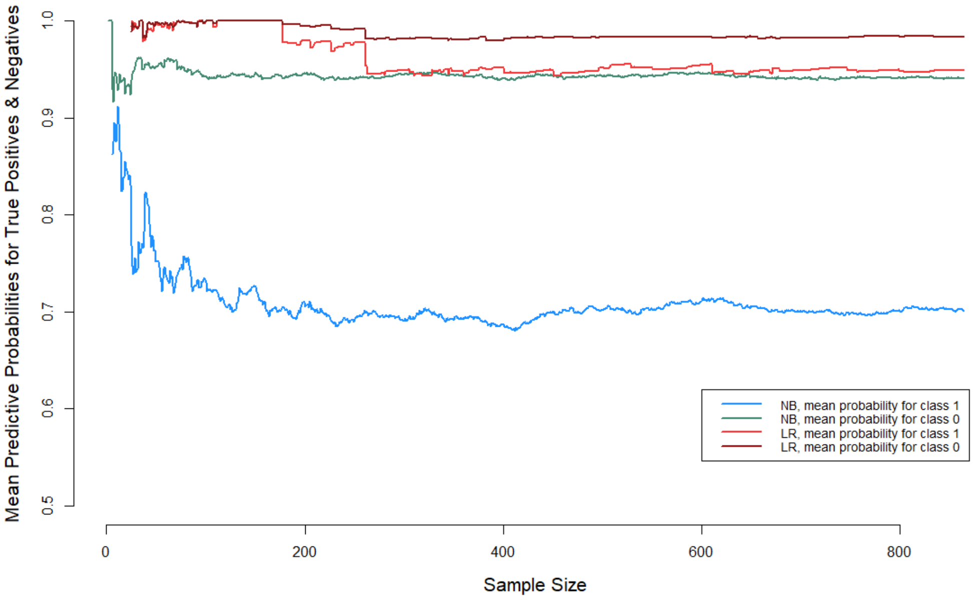

Figure 4: Mean predictive probability of each class -given that instances were correctly classified- versus ascending sample size.

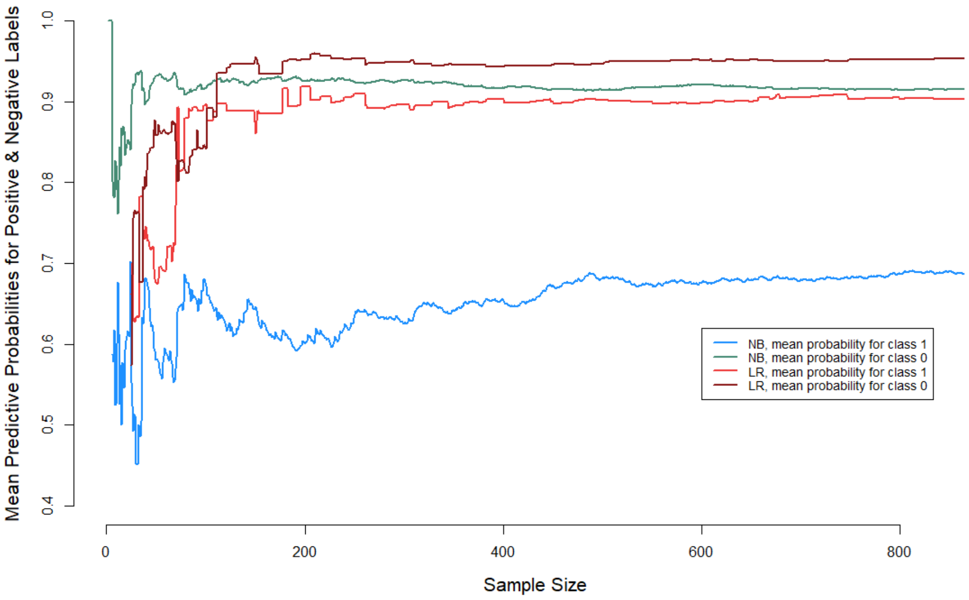

Figure 5: Mean predictive probability of each class versus the sample size.

VI. Conclusions

In terms of this study, we investigated on the performances of a Generative-Discriminative pair of classifiers, namely, Naive Bayes and Logistic Regression. Various alternations of the aforementioned classifiers were considered. Initially, we provided the theoretical background upon which, these two methods compute predictive probabilities. Subsequently, we benchmarked the pair of classifiers on a discrete feature space, for an ascending sample size. Finally, we illustrated the results, and elaborated on the strength and the weaknesses of each method, as described in Section 5.

In accordance to Andrew Y. Ng et al., NB achieves convergence faster, in $O(logn)$ whereas LR needs $O(n)$ training examples. This statement is clearly proved in their paper. Moreover, we validated that, upon convergence, LR reaches asymptotically lower error rates. Evidence on their paper is not fully clear though, as sample sizes of data sets presented are not adequate for the parameters to approximate their "true" values. Futhermore, we showed that LR yields extreme predictive probabilities due to its link function properties. Separation between the classes is, however, balanced whereas NB tends to be biased towards the Negative (majority) class.