A Comparison of Logistic Regression Vs. LDA for Wine Quality and Breast Cancer Detection

Julien Verecken, Srikanth Amudala and Kamal Maanicshah
Email: julien.verecken@mail.mcgill.ca, srikanth.amudala@mail.mcgill.ca; kamal.mathinhenry@mail.mcgill.ca

***Abstract*—This write-up discusses the implementation and performances of logistic regression and linear discriminant analysis models based on the results obtained by applying the models on a wine quality dataset and a breast cancer dataset, using 5-fold cross validation. Multiple feature subset are used to train the model, based on statistical analysis and regularization of the models, a detailed comparison is provided. Our experiments show that logistic regression is a better model when compared to LDA when it comes to accuracy. However, in time complexity LDA beats logistic regression. Both the models tend to have a low false positive and false negative rate which is especially good for medical diagnosis. Logistic Regression was able to attain 76.6% accuracy with the red wine dataset and 96.92% with the breast cancer dataset whereas LDA was able to achieve 76.1% and 95.02% respectively with a better efficiency in terms of time.**

I. Introduction

In the past decade, the field of machine learning has gotten great attention not only because of its results in fundamental research but also as a result of the applications implemented in industry. Data scientists attempt to find patterns in data and make predictions. Data mining and machine learning has become a formidable tool within the field of medical research. By the means of data mining techniques, researchers have been able to uncover new information and insights into specific and significant medical areas. Classification models have been used in a wide range of applications such as disease diagnosis, fault tolerance testing, packaging, etc. Regression models are used to uncover relationships between an outcome and response or the cause and effect of one variable over another. Linear Discriminant Analysis (LDA) is another commonly used technique for data classification and dimensionality reduction.

In this project we first implement logistic regression and then use a linear discriminant analysis (LDA) model for classification tasks. With regards to notation, we consider n samples in a dataset $X = [x_1, x_2, ..., x_n]^T$ with each sample $x_i = \left[1, x_{i1}, x_{i2}, ..., x_{i(m-1)}\right]^T$ where m is the number of features. We consider the bias term of the linear model included in the features. Furthermore, the dataset is subdivided into training and validation set which are normalized before training (zero mean, unit variance) separately to prevent any information leakage. The targeted output related to sample x_i is written as y_i.

A. Logistic Regression

Logistic regression is one of the basic statistical models used for classification tasks since it is a linear model with a number of parameters equal to the dimensionality of the dataset. This model is trained using a gradient descent scheme and models the log-odds ratio of the two classes with a linear function, this allows to add a non-linearity to the model and explore a different hypothesis space than a simple linear regression. The gradient descent method involves iteratively computing the parameters of the logistic model based on a loss function. Logistic regression minimizes the cross-entropy loss function, which is equivalent to maximizing the likelihood function of the observed samples. The gradient descent iteration is given by Eq. (1).

$$w_{k+1} = w_k + \alpha_k \sum_{i=1}^{n} x_i \left(y_i - \sigma\left(w_k^T x_i\right)\right) \tag{1}$$

$$\text{with} \quad \sigma(x) = \frac{1}{1 + \exp\left(-x\right)}$$

where $w_k = [w_{k0}, w_{k2}, ..., w_{k(m-1)}]$ represent the parameters of the linear model, α_k is the learning rate of the model. Multiple learning rate update techniques are explored in experiment III-A1 to achieve an optimal accuracy. Also, experiment III-A2 further analyses a combination of feature interactions to improve the results. The stopping criterion of the gradient descent is determined by a threshold given by Eq. 2 where ϵ is set to 10^{-4} and a maximum number of iteration is 500.

$$\|w_{k+1} - w_k\|_2 < \epsilon \tag{2}$$

The decision of the model is given after training by Eq. 3.

$$y_{\text{pred}} = \sigma(w^T x) > 0.5 \tag{3}$$

B. LDA

LDA is one of the generative learning methods used for classification. Considering a binary case ie. a dataset with 2 classes, LDA classifies new instances into the respective class based on the calculated log-odds ratio. The parameters for this classification is obtained from the training data. The formula for logg-odds ratio assuming the underlying distribution of the dataset to be Gaussian is given by Eq. (4),

$$\begin{aligned}\log \frac{P\big(y=1 \mid x\big)}{P\big(y=0 \mid x\big)} &= \log \frac{P\big(x \mid y=1\big)P\big(y=1\big)}{P\big(x \mid y=0\big)P\big(y=0\big)} \\ &= \log \frac{P\big(y=1\big)}{P\big(y=0\big)} - \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 \\ &\quad + \frac{1}{2}\mu_0^T \Sigma^{-1} \mu_0 + x^T \Sigma^{-1}\big(\mu_1 - \mu_0\big) \end{aligned} \tag{4}$$

where $P(y=0) = N_0/(N_0+N_1)$ and $P(y=1) = N_1/(N_0+N_1)$ provided N_0 and N_1 are the number of samples in the respective class and the value of μ_0, μ_1 and Σ is calculated with equations (5) and (6).

$$\mu_0 = \sum_{i=1} I(y_i = 0)x_i/N_0 \quad \text{and} \quad \mu_1 = \sum_{i=1} I(y_i = 1)x_i/N_1 \tag{5}$$

$$\Sigma = \sum_{k=0:1} \sum_{i=1:n} I(y_i = k)(x_i - \mu_k)(x_i - \mu_k)^T/(N_0 + N_1 - 2), \tag{6}$$

where μ_0 and μ_1 are the means of the respective classes and Σ is a co-variance matrix and is assumed the same for both the classes. The algorithm for classifying new instances can thus be written as:

Algorithm

1) Calculate the values of $P(Y = 0), P(Y = 0), \mu_0, \mu_1$ and Σ
2) For every x_i in the test set calculate the logg-odds ratio given by equation 4.
3) If logg-odds ratio > 0: Predict class 1
4) Else: Predict class 0

II. Datasets

The linear models are trained on two unrelated datasets : red wine's quality and breast cancer detection, to achieve binary classification. In this section, we detail the pre-processing required on each one of these datasets and investigate their statistical properties to acquire intuition about important features. However, it is essential to note the linear models that are applied to these datasets are evaluated on the accuracy metric, which is not the best representation of a model's performances since false positives and false negatives could result in different impacts. This is especially the case when working with medical data such as cancer prediction as one case is catastrophic compared to the other.

A. Red wine quality

The red wine dataset contains 11 features that takes positive real values as shown in Fig. 1.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Fig. 1: Preview of the red wine dataset

The "quality" target ranges from 0 to 10. The thresholding operation described in Eq. (7) maps the multi-class target to a binary value. Class "1" representing "good" wine class and similarly the other as "bad".

$$y_{\text{binary}} = y_{\text{multiclass}} > 5 \tag{7}$$

The histograms specific to each feature, separated by class (Fig. 2) allows to perform a first analysis. A trace is added on top of histograms using an automated kernel density estimation. We can observe all features have a non symmetric distribution since they can only take positive values. We already perceive some features allow an easy discrimination between the two classes since their distributions have a limited overlap. This is the case for "volatile acidity","density","sulphates" and "alcohol".

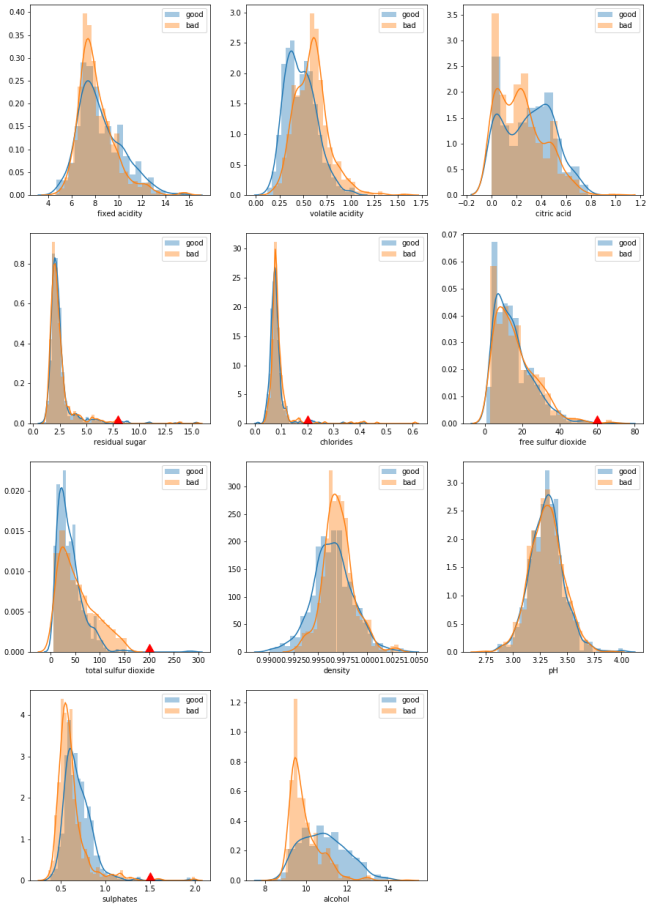


Fig. 2: Red wine histograms by feature and class

Also, several features have heavy skewed distributions where some values could be considered as outliers. We chose to eliminate those problematic samples that could have a negative impact in the training of our models. The red triangles in Fig. 2 indicates the thresholding operation that is performed to delete high value samples. This filtering operation removes around 10% of the total dataset equally among both classes. The statistics of the dataset, by class, is computed in Fig. 3 and reveal an unequal balance between the two classes (53.5% good wines) and the mean of the 4 features mentioned above are the most separated. The correlation coefficients of all features and the target are displayed in Fig. 4. Since it is the case chemically, all measurements related to acidity have a strong correlation coefficient. Separately, we observe the same behavior for sulfur measurements. Regarding the target variable "alcohol", "volatile acidty" and "sulphates" have the largest linear relationship.

		fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	bad	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.000000	712.0
	good	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.000000	821.0
mean	bad	8.135815	0.590913	0.229719	2.462500	0.084289	16.289326	53.942416	0.997033	3.320070	0.599916	9.949017	0.0
	good	8.485384	0.474629	0.295859	2.394945	0.078210	15.080999	38.140073	0.996418	3.315006	0.687101	10.883739	1.0
std	bad	1.586386	0.178663	0.177253	1.033537	0.020169	10.369163	36.707920	0.001576	0.149077	0.128494	0.759135	0.0
	good	1.880440	0.162936	0.199731	0.874236	0.019639	9.631748	23.297621	0.002030	0.153054	0.136344	1.097332	0.0
min	bad	4.600000	0.180000	0.000000	1.200000	0.039000	3.000000	6.000000	0.992560	2.880000	0.330000	8.500000	0.0
	good	4.700000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.860000	0.390000	8.400000	1.0
25%	bad	7.100000	0.460000	0.080000	1.900000	0.074000	8.000000	23.000000	0.996100	3.220000	0.520000	9.400000	0.0
	good	7.100000	0.350000	0.100000	1.900000	0.066000	7.000000	21.000000	0.995160	3.220000	0.590000	10.000000	1.0
50%	bad	7.800000	0.590000	0.215000	2.200000	0.081000	14.000000	44.000000	0.996900	3.310000	0.570000	9.700000	0.0
	good	8.000000	0.460000	0.310000	2.200000	0.077000	13.000000	33.000000	0.996400	3.310000	0.660000	10.800000	1.0
75%	bad	8.900000	0.680000	0.340000	2.600000	0.092000	23.000000	77.000000	0.997900	3.410000	0.640000	10.300000	0.0
	good	9.700000	0.580000	0.460000	2.600000	0.086000	20.000000	49.000000	0.997600	3.400000	0.770000	11.700000	1.0
max	bad	15.900000	1.580000	0.790000	7.900000	0.186000	57.000000	155.000000	1.003150	3.900000	1.220000	14.900000	0.0
	good	15.600000	1.040000	0.760000	6.700000	0.194000	54.000000	165.000000	1.003200	4.010000	1.360000	14.000000	1.0

Fig. 3: Red wine statistics

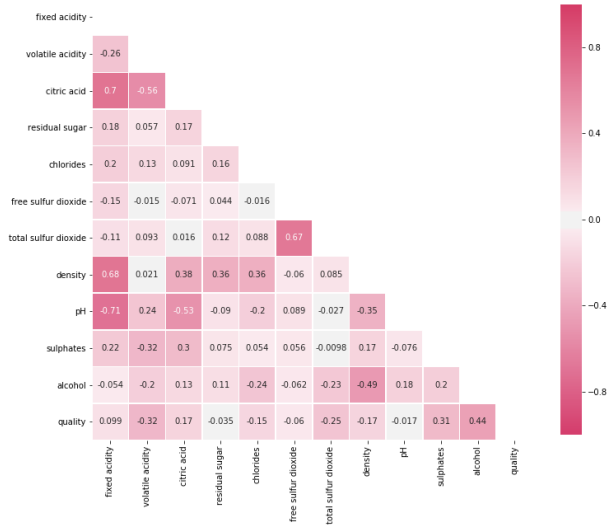


Fig. 4: Red wine correlation coefficients

This dataset exhibits a complicated structure with many features having an overlapping distribution and we expect limited accuracy with a simple linear model. In Section III, interaction terms of any pair of feature are generated in the aim of further obtaining distinguishable distributions.

B. Breast cancer

The breast cancer dataset contains 9 features that takes integer values ranging from 1 to 10 as shown in Fig. 5 and a "Sample Code Number" that we immediately discard for the learning process. The two classes, that takes value 2 and 4 respectively, are converted to a "benign" class 0 and a "malignant" class 1. It is important to note that 16 samples contain missing value for the "Bare Nuclei" feature and are consequently being deleted. The histograms displayed in Fig. 6 shows all features have distinguishable distribution, with the exception of "Clump Thickness". Fig. 7 shows the mean separation between the benign and malignant class and a more concentrated benign class. The "benign" class is also more dominant in the dataset (65%), that will constitute our baseline to evaluate the model performances. The correlation matrix in Fig. 8 indicates features that are highly correlated with each other with the exception of "mitoses", "Uniformity of Cell Size" and "Uniformity of Cell Shape" have the strongest linear relation and are practically identical to each other.

Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
776715	3	1	1	1	3	2	1	1	1	2
841769	2	1	1	1	2	1	1	1	1	2
888820	5	10	10	3	7	3	8	10	2	4
897471	4	8	6	4	3	4	10	6	1	4
897471	4	8	8	5	4	5	10	4	1	4

Fig. 5: Preview of the breast cancer dataset

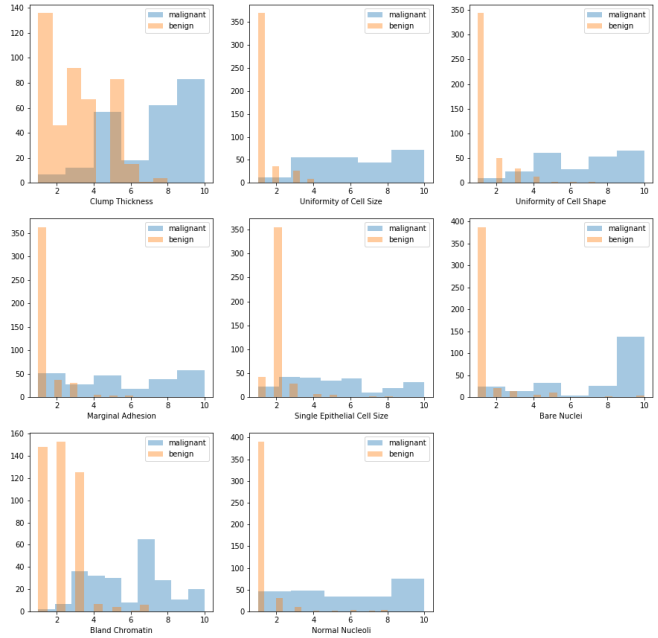


Fig. 6: Breast cancer histograms by feature and class

		Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
count	benign	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	444.000000	444.0
	malignant	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.0
mean	benign	2.963964	1.306306	1.414414	1.346847	2.108108	1.346847	2.083333	1.261261	1.065315	0.0
	malignant	7.188285	6.577406	6.560669	5.585774	5.326360	7.627615	5.974895	5.857741	2.602510	1.0
std	benign	1.672661	0.855657	0.957031	0.917088	0.877112	1.177848	1.062299	0.954606	0.509738	0.0
	malignant	2.437907	2.724244	2.569104	3.196631	2.443087	3.116679	2.282422	3.348876	2.564495	0.0
min	benign	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.0
	malignant	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0
25%	benign	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	1.000000	1.000000	1.000000	0.0
	malignant	5.000000	4.000000	4.000000	3.000000	3.000000	5.000000	4.000000	3.000000	1.000000	1.0
50%	benign	3.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	1.000000	0.0
	malignant	8.000000	6.000000	6.000000	5.000000	5.000000	10.000000	7.000000	6.000000	1.000000	1.0
75%	benign	4.000000	1.000000	1.000000	1.000000	2.000000	1.000000	3.000000	1.000000	1.000000	0.0
	malignant	10.000000	10.000000	9.000000	8.000000	6.500000	10.000000	7.000000	9.500000	3.000000	1.0
max	benign	8.000000	9.000000	8.000000	10.000000	10.000000	10.000000	7.000000	8.000000	8.000000	0.0
	malignant	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	1.0

Fig. 7: Breast cancer statistics

III. Results

A. Logistic regression

1) Learning rate for the logistic regression: Recalling Eq. (1), with the gradient descent of the logistic regression algorithm, we find the minimum of the cross-entropy loss function. This being a convex function, a local minimum is also a global minimum and it is unique. Hence, the gradient descent scheme in this context can begin with the same starting point. We

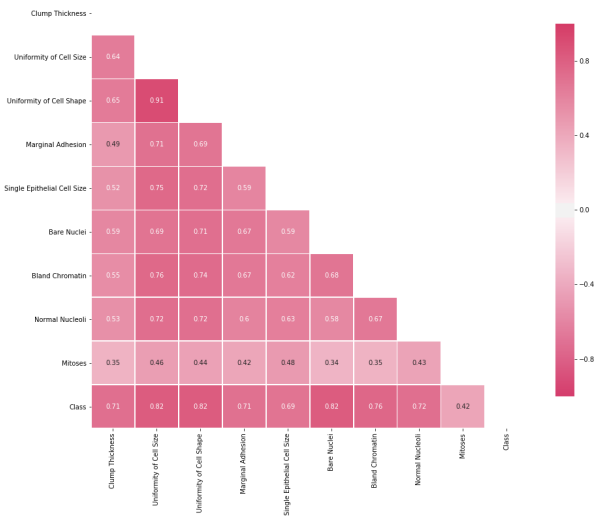


Fig. 8: Breast cancer correlation coefficients

choose to start from the zero-vector and now the step size have to be determined. The "Robbins-Monroe" condition (Eq. (8)) establishes sufficient conditions to ensure convergence of w_k to a local minimum of the error function.

$$\sum_k \alpha_k = \infty \quad \text{and} \quad \sum_k \alpha_k^2 < \infty \tag{8}$$

Starting from the function that satisfies this condition $\alpha_k = 1/(k+1)$, we define a new parametric function (Eq. (9))

$$\alpha_k = \frac{k}{k+\beta}\alpha_{k-1} \quad \text{and} \quad \alpha_0 = \alpha_{\text{init}}. \tag{9}$$

which is an hyperbolic function that allows to tune the starting point as well as the decay rate, such as shown in Fig. 9. Observe that for $\beta = 0$, we have a constant learning rate.

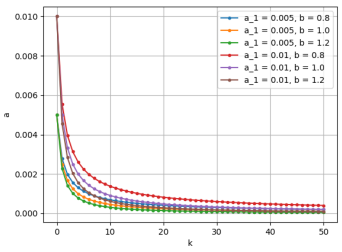


Fig. 9: α_k function

The following experiment involves finding the best pair $(\alpha_{\text{init}}, \beta)$ by 5-fold cross validation with other hyper-parameters fixed. The maximum number of iterations is set to 500 while the threshold ϵ to detect the convergence is 10^{-4}.

The results displayed in Fig. 10 shows an optimized pair $(\alpha_{\text{init}}, \beta) = (0.01, 1)$, although all combinations give very similar results. Furthermore we can check the learning curves for these optimized parameters as a function of the number of iterations. Figure 11a shows a fast convergence from the initial

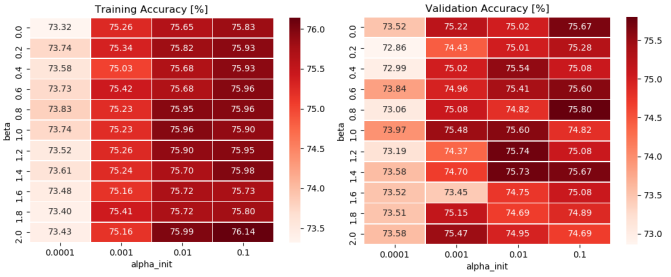


Fig. 10: Results of experiment III-A1

point towards the maximum accuracy in less then 40 iterations. The final accuracy on the validation set is over 75% which is more than 20% improvement over the baseline.

A similar experiment with the breast cancer dataset gives an ideal learning pair of $(\alpha_{\text{init}}, \beta) = (0.001, 0.8)$, resulting in an accuracy over 97% on the validation set. This confirms the intuitions we had in the dataset analysis, where the classes of breast cancer are more easily separable.

2) Feature augmentation and selection: This second experiment tries to improve the accuracy on the red wine dataset by adding features that are square of features as well as interaction of a pair of two initial features of the dataset (Eq. (10)).

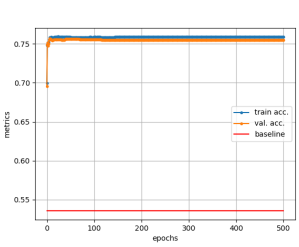
$$x_{ij} = x_i * x_j \quad \text{with} \quad 0 \leq i \leq j \leq m \tag{10}$$

Including the bias term, there are 78 features in total which makes the algorithm less efficient and difficult to tune. A similar optimization as in experiment III-A1 results the following optimal parameters $(\alpha_{\text{init}}, \beta) = (0.005, 0.2)$, and gives a training accuracy over 77% and a validation accuracy around 76%. A solution to decrease complexity is to use "L1" regularization on the model to optimally select a few number of features without decreasing the final accuracy. We use a large value for the regularization factor λ in order to select a small subset of the initial features. The parameters of the experiment are the same as previously mentioned for the stopping criterion. Additionally, we have parameters $\lambda = 5$, $\alpha_{\text{init}} = 0.005$ and $\beta = 0.2$.

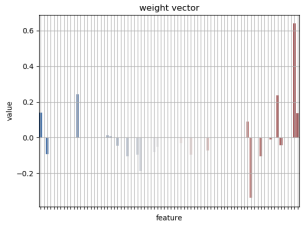
The resulting weight vector is shown in Fig. 11b, it is clear "L1" regularization has pushed down a lot of parameters and we are left with a dozen of mixed features of great interest. As the features are sorted in a similar fashion as a correlation matrix, we can observe that features and interaction features linked with "alcohol", "sulphates", "volatile acidity" and "density" as well as the bias term have a great impact in the weight vector. We can therefore try to train a model solely on these new features. A summary of the results is reported in Table I, the numbering refers to the dataset feature ordering (counting from 0) (see Fig. 1). To conclude, there is no real advantage of including cross terms into the model. As we induced in Section II-A, alcohol is the most important when ranking features.

Features	Accuracy(%)
10 most important features	75.9%
4 most important features	73.4%
1 most important features	71.1%
Original features	75.3%
[1,7,9,10]	73.4%
[10]	70.4%

TABLE I: Comparison results of interaction terms



(a) Learning curve

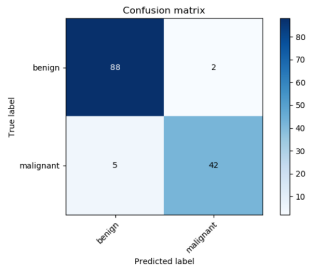


(b) Weight vector result of experiment III-A2

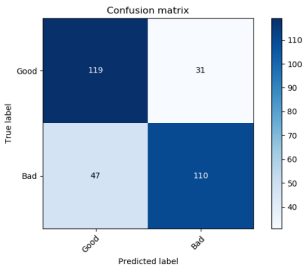
Fig. 11: Parameter Tuning

B. LDA

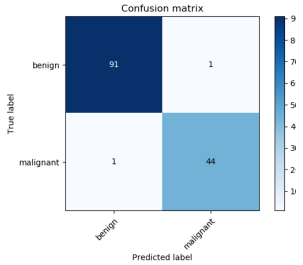
Regarding the red wine dataset, the goal is to classify the quality of the wine, with 0 being bad and 1 being good. Using Eq. (5), Eq. (6), μ, Σ, $P(y=0)$, $P(y=1)$ are estimated using the training data and using Eq. (4) the log-odds ratio for prediction can be computed which leads to an accuracy of 76.1%. On the breast cancer dataset, the goal was to predict if a tumor is benign or malignant and we are able to achieve an accuracy of 95%.



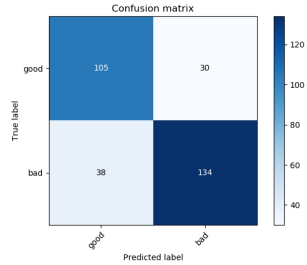
(a) Breast cancer for LDA



(b) Red wine for LDA



(c) Breast cancer for LR



(d) Red wine for LR

Fig. 12: Confusion Matrix for the two datasets

C. Complexity and accuracy comparison

In our experiments we observed a time complexity as shown in table II. We see that the logistic regression model consumes more time to execute. The confusion matrix of the predictions for both the models and datasets are shown in Fig. 12 which exhibits the performances of our model with unbalances towards false "benign" and false "good".

Model	Dataset	time(s)	Accuracy
Logistic Regression	Red Wine	6.67	76.6%
Logistic Regression	Breast Cancer	9.46	96.92%
LDA	Red Wine	0.025	76.1%
LDA	Breast Cancer	0.021	95.02%

TABLE II: Comparison results of time taken for the models to execute

IV. Discussion

For the wine dataset, the time complexity is much lower in LDA when compared to logistic regression because the regression algorithm has to run through a number of iterations before convergence whereas the parameters of the LDA model can be directly calculated. However, when we look at the accuracy for the two models logistic regression scores better than the LDA model. It is to be noted that the accuracy mentioned are for the respective validation sets of the two datasets. From these results it is evident that there has to be a trade-off between time and accuracy. Both the models avoid the over-fitting problem as the training accuracy was around the similar range. It is found from the confusion matrices that the false positives and false negatives for both the models are low. Especially, for the breast cancer dataset where the false positives and false negatives has to be low the model performs very well.

V. Conclusion

The work presents a detailed comparison of logistic regression and LDA. We have used two challenging datasets (wine quality and breast cancer) to evaluate the performance of our models. The performance metrics show that logistic regression performs better than LDA. Howevever, the time trade-off between the two models is pretty high. Hence, it would be a better call to use LDA in case of large datasets where time is a constraint. In case of small datasets, logistic regression would be a better choice.

VI. Statement of Contributions

Julien Verecken, has done dataset analysis, implemented and made documentation for Logistic Regression and kfold cross validation. Srikanth Amudala and Kamal Maanicshah has implemented LDA. All the three made great efforts in terms of writing and implementation.