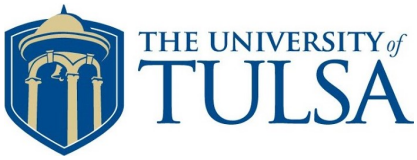



Deep Learning
CS-6863

# Homework 3

*Instructor:*
Mahdi Khodayar

*Author:*
Selim Karaoglu

February 16, 2022

In this homework assignment there are two tasks two achieve; first task is to choose a dataset provided by UCI ML Repository and implement Logistic Regression and Gaussian Naive Bayes for this dataset to make comparison between these methods, following this, second task is to implement regularization for Logistic regression and make comparison between Logistic Regression model with and without parameter regularization on the same dataset.

# Introduction

This project is designed to implement Logistic Regression (LR) and Gaussian Naive Bayes (GNB) classification methods to solve the problem on edibility of mushrooms. The mushroom dataset is provided by the UCI ML Repository[1]. Methods section provides in-depth explanation about the classification methods and their implementation, in addition to that, information about the mushroom dataset is provided in this section. Experiment section focuses on the application of the classification data and the results. The comparison of the classification models' accuracies are presented in the Experiment section.

# Methods

This section highlights the LR and GNB classification models, also explains the dataset with the details about it's features. First, the classification models are explained with the mathematical principals they constructed on, further both algorithms are trained and tested on the mushroom dataset. The accuracy scores for both classification models are compared to see which method has the higher score, therefore is a better choice for this dataset.Finally, the mushroom dataset is presented with its features. The structure of the dataset is explained further in detail with the features as substructures, correlations between features, distribution of the values for each feature etc.

## Logistic Regression

Logistic Regression is one of the classification methods that are widely employed in the machine learning projects. Logistic Regression is used to predict a binary output given a set of independent variables. This suggest that there are only two possible predictions for this classification algorithm; the possibility of event happening (1) and not happening (0). In addition to that, Logistic Regression method favors independent variables, in other words, features should be independent from each other. High correlation between features suggests that the features are correlated and therefore not independent from each other.

Logistic Regression maps the output to a binary classification, therefore output can only be 0 or 1. To map the outputs, LR uses the sigmoid function; that creates a smooth curve between 0 and 1:

$$f(x) = \frac{1}{1 + e^{-x}}$$

By using the sigmoid function, LR method maps the output between 0 and 1. After the prediction, it is possible to compare the predictions and the actual values. Loss function is utilized to calculate the error between the predicted values and actual values. Log-loss function is the method embraced by the Logistic Regression method and can be defined as:

$$LogLoss(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)), & \text{if y=1.} \\ -log(1 - h_\theta(x)), & \text{if y=0.} \end{cases} \tag{1}$$

This Log-loss function can be simplified to:

$$Logloss(h_\theta(x)) = -ylog(h_\theta(x)) - (1 - y)log(1 - h_\theta(x))$$

With this Log-loss function, it is possible to calculate the loss for the single $h_\theta(x), y$. For the whole dataset, the loss function can be defined as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))]$$

For the L1 parameter regularization, Gradient Descent method formula needs the addition of $(sum(|w|))\alpha/m$ and for the L2 parameter regularization the addition needs to be $(sum(|w|)^2)\alpha/m$. Loss function is adjusted likewise.

[1] https://archive.ics.uci.edu/ml/datasets/mushroom

Another important algorithm adopted by Logistic Regression is the Gradient Descent method. The Gradient Descent method is used to find optimal feature values. This formula takes learning rate into the calculation, the size of the steps the algorithm needs to take the find the optimum solution. As the definition suggests, the function runs until it converges. The Gradient Descent method can be defined as:

$$repeat\ until\ convergence \begin{cases} \theta_0 = \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\ \theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)} \end{cases} \tag{2}$$

Logistic Regression implementation for this work follows this algorithm:

**Algorithm 1** Logistic Regression

**procedure** LOGISTICREGRESSION
- **function** SIGMOID
  calculate the sigmoid with given "z" as: $1/1 + e^{(-z)}$
- **function** LOSS
  calculates the log-loss for Gradient Descent, L1 and L2 regularization methods.
- **function** ADD_INTERCEPT
  creates interception point for features matrix.
- **function** FIT
  trains the model on the given dataset.
- **function** PREDICT_PROB
  returns the predictions for the given data.
- **function** PREDICT
  calls predict_prob function and returns the rounded output.

## Naive Bayes

Naive Bayes is a machine learning algorithm, more specifically, it is a classification method. This means that Naive Bayes is used when the output variable is discrete. The underlying methods of the algorithm are based on the Bayes Theorem that states:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

To make it easier to understand, this equation can be written using the features (attributes, input values) and the targets (output values). In simpler terms; this function can be employed to solve for the probability of $y$ with given features $X$ as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

The Naive assumption takes place here on the formula. Naive Bayes method works on cases where different feature values are independent from each other. Therefore the formula can be rewritten as:

$$P(X|y) = P(X_1|y)P(X_2|y)...P(X_n|y)$$

It is clear that the goal is to solve for $y$, in this circumstance P(X) is a constant value and it can be removed from the equation. In addition to that, by implementing the proportional nature of the $X_1, X_2, ..., X_n$ the equation can be presented as:

$$P(y|X) = P(y) \prod_{i=1}^{n} P(X_i|y)$$

Since the values across the datapoints can be calculated, the last step should be the finding the maximum of these calculated output values of $y$. The maximum can be obtained with the arg max function:

$$y = \arg\max_y [P(y) \prod_{i=1}^{n} P(X_i|y)]$$

With taking consideration on provided information about Naive Bayes, the classification method can be explained as; first, create a frequency table and then a ratio table so that the values for P(X) and P(y|X) can be calculated, furthermore, for a given set of input features X, compute the proportionality of P(y|X) for each class y. In this example, the implementation should be binary or simply True or False output. Now as the last important part of the Naive Bayes method, there could be different assumptions when calculating $P(X|y)$ like Multinomial model, Bernoulli model and Gaussian model. This implementation adopts the Gaussian model with the assumption on the distribution of $P(X_i|y)$ follows a normal (Gaussian) distribution. Gaussian Naive Bayes model is based on the assumption of continuous values are sampled from a normal distribution, this assumption can be shown as:

$$P(X_i|y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

Naive Bayes implementation for this work follows this algorithm:

**Algorithm 2** Gaussian Naive Bayes

**procedure** NAIVEBAYES

**function** PRIOR

calculate prior probabilities P(y)

**function** STATS

calculates mean ($\mu$) and variance ($\sigma$).

**function** DENSITY

calculates the probability from the Gaussian Distribution. $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$

**function** POSTERIOR

calculates the posterior probability for each class and returns the class with the highest posterior probability.

**function** FIT

trains the model on the given dataset.

**function** PREDICT

returns the predictions for the given data.

## Dataset

This section provides an insight about mushroom dataset. The authors of this dataset defined this as: "This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like 'leaflets three, let it be' for Poisonous Oak and Ivy". This dataset contains 22 features about mushrooms and an output that represents the edibility of the mushroom, 1 for edible and 0 for inedible. All 22 features of the mushroom dataset are discrete valued. This discrete valued structure is favorable for GNB algorithm. However, there is lot more details about this dataset that might put some light on the accuracy of the classification methods. Distribution of each feature plays an important role on the accuracy of the classifiers, in addition to that, Independency of the features is one of the most important aspects on model selection.

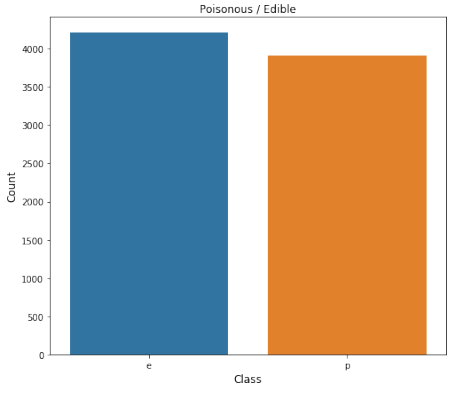



Figure 1: Output values distribution for Mushroom Dataset

The distribution of the features might provide useful information about the dataset. On a machine learning project, the classification methods are selected by the accuracies they achieve and this accuracy is dependent on the distribution of each feature. Gaussian Naive Bayes algorithm assumes that values are distributed with normal distribution for features, therefore if the dataset contains distributions that does not follow

normal distribution, the accuracy of GNB method is expected to suffer. Logistic Regression model is not effected by the distribution of the features since it is not a probabilistic approach. Figure 1 shows the distribution of the output values. There are 8124 datapoints in this dataset, 4208 of them are labeled as edible and 3916 of them are inedible. The outputs values are close to each other well enough to not to have any problems on training and test splits.

In Figure 2, the distribution for the values of each feature that has more than 2 values is shown. There are 16 features shown in the Figure 2. As can be seen, while some of the feature distribution follows a distribution that is similar to normal distribution (for example stock colors above and beyond rings), most of the features are not distributed normally. This poses a big challenge for the GNB method.

Beside of following the normal distribution, this figure provides some other valuable information about the dataset; there are some features that are distributed in very imbalanced shape.

- Figure 2 shows that features called "veil-color" and "ring-number" contains one big spike, that suggest that most of the datapoints have the same value and few others have different values. This might cause an issue since the training and test splits on dataset are performed randomly and this randomness might result with the training set to not contain any other value than the spiked value, therefore might have problems on classification when introduced a feature value it has never seen.
- These features can be overlooked (by simply removing them) or each split can be designed to contain representative number of each feature (randomness does not apply anymore in that case) etc.
- It is possible to overcome this problem and obtain better results with the classifiers, however this is out of the scope of this project and therefore suggested as a future work.
- The best option would be the selecting the model for the dataset but this project focuses on evaluating the LR and GNB methods, therefore selecting a classification model that would make classification more accurate is out of context of this work and therefore suggested as a future work.

The mushroom dataset has balanced output values but imbalanced feature values. The feature values are not favorable by GNB owing to the fact that most of them are not distributed normally. Some of the features are spiked with one value, which is another obstacle that classification methods needs to face.

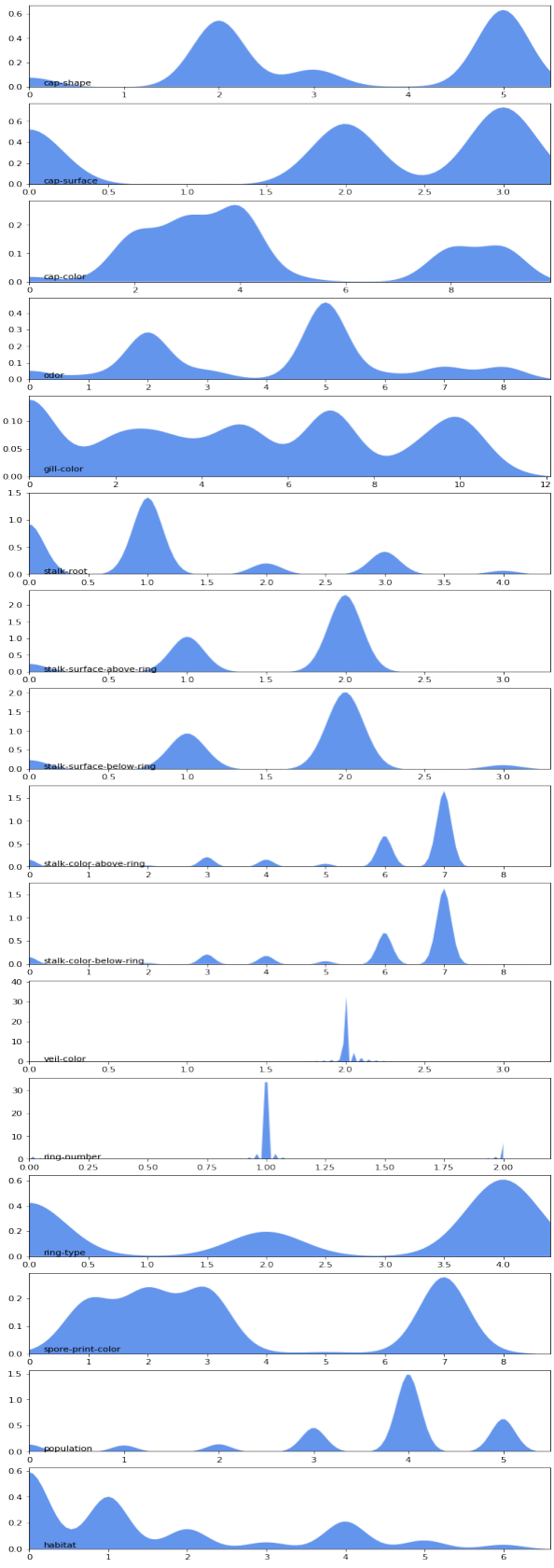



Figure 2: Feature values distribution for Mushroom Dataset (i>2)

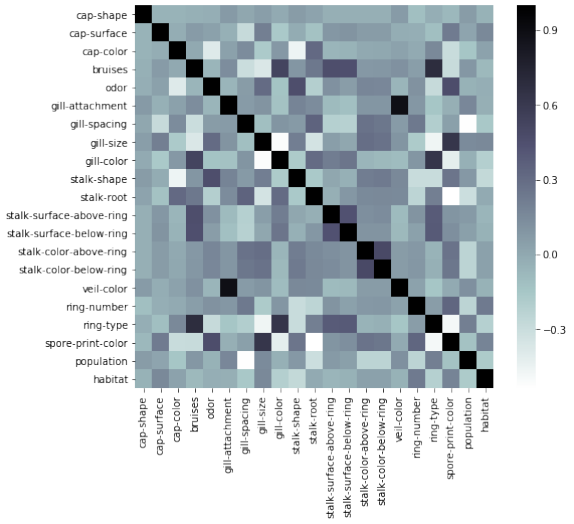
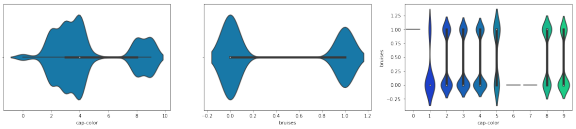
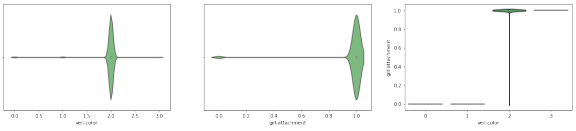


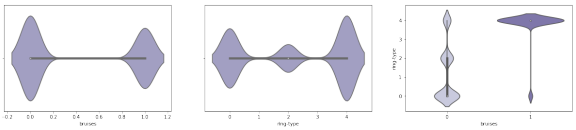

Figure 3: Feature values correlation heat map for Mushroom Dataset



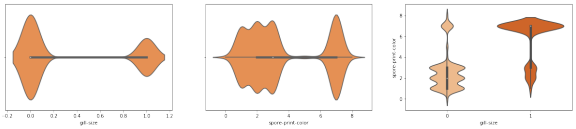
(a) Cap color vs Bruises. C = -0.000764



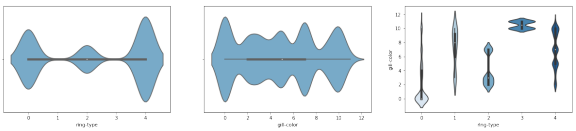
(b) Veil color vs. Gill attachment. C = 0.897518



(c) Bruises vs. Ring type. C = 0.692973



(d) Gill size vs Spore print color. C = 0.622991



(e) Gill color vs. Ring type. C = 0.629398

Figure 4: Distribution plots of the correlated features

Another important aspect of the dataset is the independence of the variables. To achieve this goal, the correlation heatmap of the features can be examined. Figure 3 shows the correlation heat map among all 22 features; the dark colored points on the plot suggest that there is a strong correlation between those features. Both LR and GNB classification methods work better if the features are independent from each other, in this case there are some highly correlated features in the mushroom dataset and this is expected to affect the accuracy for both LR and GNB in a negative direction. Figure 3 suggests there are strong correlations between;

- "gill attachment" and "veil color" with 0.897518
- "bruises" and "ring type" with 0.692973
- "gill size" and "spore print color" with 0.622991
- "gill color" and "ring type" with 0.629398

These correlations can be seen in detail with the distribution plots for each feature and distribution plots for combination of these correlated features. All of the subfigures in Figure 4 except Figure 4.a provides the feature distribution plots for features that are highly correlated. Figure 4.a shows two of the least correlated features; "cap color" and "bruises" with -0.000764 correlation. The first plot shows the distribution for feature "cap color" and the plot suggest that the values are distributed in a similar fashion to normal distribution. Second plot of Figure 4.a presents the value distribution for feature "bruises" and the distribution is well enough to not to cause issues on training. Third plot of Figure 4.a shows the distribution type that LR and GNB models favor. As the plot suggests, the data is distributed with normal distribution and the features are independent from each other. The Figure 4.a represents the ideal feature correlation expectation of the dataset for LR and GNB methods. However, the remaining subfigures from the Figure 4 shows that the dataset is far from being fit for this expectation. Figure 4.b suggests that the correlation between "veil color" and "gill attachment" features are very high, therefore these features are not independent from each other, in addition to that, the values of both features are not distributed closely, one of the values contain most of the datapoints. Figure 4.c, 4.d and 4.e supports the idea presented for Figure 4.b, this inclines that the mushroom dataset possibly might be learned and classified better with other machine learning methods with different assumptions of the dataset. The LR and GNB methods might result if the dataset is refined without these highly correlated features, however this would be out of context for this project and might be focused on future works.

# Experiment

This section focuses on training the LR and GNB methods on the mushroom dataset and presenting the accuracy scores for both classifiers. Further, L1 and L2 parameter regularization is applied for the LR method and the results are compared. The experiment results are presented in two subsections to point out the answers for the questions presented in the homework.

## Question 1

Choose one of these problems/datasets, and implement Logistic Regression and Gaussian Naïve Bayes for that problem. Is GNB better than LR?, or LR is better? why?

The mushroom dataset is selected as a dataset and the implementations for LR and GNB models are explained above with details. The only part that is unanswered is which classification method performed better. GNB model does not take any parameters but LR model takes "learning rate", "lambda" and "epoch size" as parameters and these might effect the accuracy of the models. Therefore to find the best parameters for the LR model, first step is to do cross validation. The cross validation applied for the LR model suggests that the best "learning rate" is 0.01 and the best "lambda" value is also 0.01. Epoch value is not included in cross validation.

After obtaining the best parameters for the LR model, it is possible to compare it with the GNB model. Both models are trained on %70 of randomly selected data from dataset and tested on the remaining data. The LR model without any parameter regularization resulted with 0.88 weighted average accuracy score while GNB only scored 0.7. For the mushroom dataset; the LR model had a higher accuracy than GNB model and therefore LR model is better than GNB for this specific dataset.

## Question 2

We saw two versions of training LR. The first one was without parameter regularization (without P(W)), and the second one was with P(W). Compare these two LR training algorithms in your dataset (defined in Question 1).

Comparing LR models on the mushroom dataset is not the best comparison. This is due to several reasons; the features are not independent in the mushroom dataset, some features have biased distributions and the features are discrete. Owing to these unfit specifications of the dataset for LR, the parameterization does not help to increase the accuracy a lot since these specifications are not representable by the parameterization of LR. Figure 5.a shows the training loss for three different LR models; LR without regularization, LR with L1 regularization and LR with L2 regularization, the figure suggests that there is almost no difference on loss through the training for all three models. LR with no regularization surprisingly performed very slighlty better than other LR models, but there is no significant difference. Figure 5.b shows the accuracies for three LR models and the GNB model. LR models outperformed the GNB model on every different training-test size splits. LR models performed very similarly on different splits. Both figures suggest that the LR algorithm parameter regularization is not effective on this specific dataset and this might be caused by the structure of the dataset.

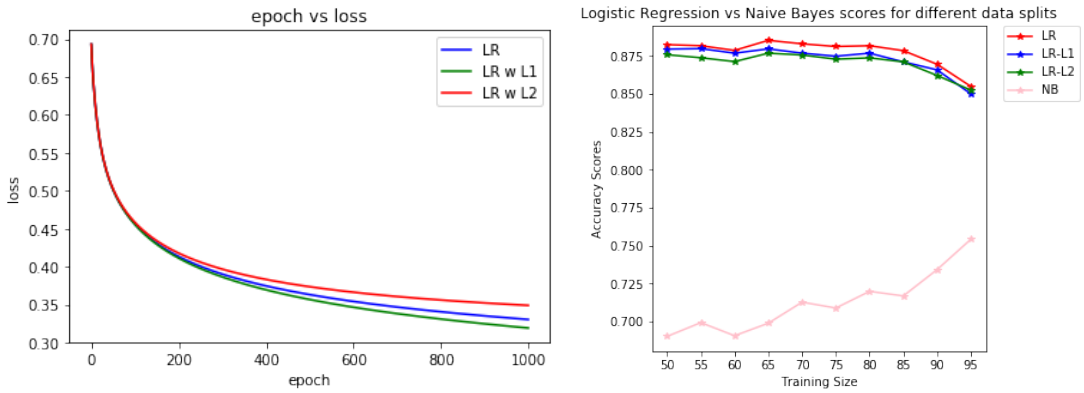



Figure 5: Training Loss and Accuracies on different split sizes

## Conclusion

Logistic Regression and Gaussian Naive Bayes classifiers are utilized to help understand the data and make classifications about it; both LR and GNB algorithms are implemented in this project to solve the mushroom edibility problem with the mushroom dataset provided by UCI ML repository. The work on the dataset showed that the features are not independent while both models favor independency between the features. Also some of the features are not distributed well enough that they might cause unseen examples problem on model testing. The LR models performed better than GNB on every test. Even though GNB favors discrete valued features, this was not enough for GNB to show a good performance. LR models also did not reach accuracies higher than %93, although the results are acceptable comparing to GNB. All models scored similarly on weighted average accuracy with 0.88. LR with L1 and L2 parameter regularizations did not affect the accuracies drastically, however; LR with L1 and L2 regularization scored slightly higher than LR without regularization.