|  |  |
| --- | --- |
| **DATA 430 Technical Report Assignment 1 (a & b): Logistic Regression** | **Kimberly Hernandez** |
| **Email Spam Classification Using Machine Learning** | |
| **URL to dataset:** <https://archive.ics.uci.edu/dataset/94/spambase> | |

**Assignment 1a (due Week 2):** you should complete the following sections ONLY:

* Overview (Problem Domain)
* Overview (Objective)
* Analysis (Exploratory Analysis)

**Assignment 1b (due Week 3):** all sections of this template should be completed. Modifications of the three sections submitted in Assignment 1a should be made based on feedback from the instructor.

This template should be used in conjunction with the assignment instructions. The size of the text area below will expand to the length of your response; the area should not be interpreted as a required or suggested length of response. Responses within the text area should be single spaced with Times New Roman 12pt font. The body of the document will likely be 6-9 pages, not including the Appendix; length may vary depending on specifics of the analysis and the dataset. As needed, APA format in-text citations should be included, along with a full references list at the end of the document.

|  |
| --- |
| **Overview** |
| **Problem Domain**: give some background and context about the problem domain (application area). For instance, if you are doing the analysis for predicting heart disease, provide some context about the disease and include some interesting statistics about it. Also, discuss how the method is relevant for the chosen problem. |
| Email spam detection is one of the most important features for mailbox websites such as Gmail, Outlook, and Yahoo Mail. These platforms serve a large number of users globally, and unwanted emails can raise serious risks (i.e., phishing attacks, fraud, and malware spread), where safe communication can become compromised. Machine learning can offer a reliable solution to this problem domain as it can learn directly from historical data and identify patterns that can distinguish whether an email can (or cannot) be labeled as spam. This supervised learning approach would include using logistic regression, as it is designed for binary classification modeling (1 = True, 0 = False), where true (1) values would confirm spam, and false (0) values would detect that an email is not spam. Furthermore, using numerical features that include the frequency of words, characters, and capital letter usage can be applied to a logistic regression model to estimate the probability that a message belongs to the spam category. |
| **Objective**: Clearly state the objective of the analysis in relation to the kind of algorithm you are employing. Use specific language as to what question(s) you are trying to answer using the specific analysis/modeling type. |
| The main objective for this analysis is to apply logistic regression in order to classify emails as spam or non-spam using the Spambase dataset from the UCI Machine Learning Repository (University of California, Irvine, 1999). The main research question for this project is: “Can Logistic Regression predict the likelihood that an email is spam based on text-related features with great accuracy?” The Logistic Regression model will specifically examine how patterns such as word frequencies, capitalization behaviors, and character frequencies can influence the likelihood of a message being classified as spam. The overall purpose of this study is to evaluate the model’s accuracy, interpretability, and suitability for spam classification, while also identifying which features are most predictive of spam content. |
| **Analysis** |
| **Exploratory Analysis**: describe the data including the source, the collection method, and variables. Perform exploratory analysis. Also, select few key variables (including the target variable for supervised learning) and study their distributions using plots such as histograms, box plot, bar chart, etc. |
| The dataset used for this model is the Spambase dataset, which was collected by researchers and made public in 1999 through the UCI Machine Learning Repository. The dataset contains 4,601 mail entries and 57 columns. Independent variables are comprised of 48-word frequency attributes, 6 character frequency attributes, and 3 attributes that measure capitalization patterns (i.e., the avg. length of consecutive capital letters, the longest run of capital letters, and the sum of capital letters in each message). The dependent variable (y label) is the binary classification, which identifies whether the email is spam (1 = True) or non-spam (0 = False).    The three boxplots above provide data visualizations of how selected features are distributed across the binary classification: Spam (1) and Non-Spam (0).  **1st Boxplot: Word Frequency**  Displays the distribution of the variable ‘word\_freq\_1” by each binary class. Most values for both spam and non-spam emails are below two and are concentrated near zero. More outliers can be seen in the non-spam distribution, with some present in the spam classification.  **2nd Boxplot: Character Frequency**  Displays the distribution of the variable ‘char\_freq\_1’ against each binary class. Similar to the 1st boxplot, most values are closer to zero, and a cluster of outliers can be seen in the non-spam category. This confirms that specific characters are more often used in spam emails and are more commonly found in low-frequency sequences.  **3rd Boxplot: Capitalization**  Provides the distribution of “capital\_run\_length\_average” against each binary class, which measures the average length of consecutive capital letters in an email. The results show that spam emails contain much higher average runs of capital letters compared to non-spam emails. This finding is consistent with the common observation that spam messages often use excessive capitalization for emphasis, which can be seen in the wide range and extreme outliers within the spam class.  **Key Insights Summary from Boxplots:**  Overall, these visualizations reveal that while most feature values are concentrated near zero, spam emails are more likely to contain higher word frequencies, unusual character usage, and excessive capitalization. These patterns suggest that logistic regression can assess the performance of these specific features to separate spam from non-spam messages effectively. |
| **Preprocessing**: armed with the exploratory analysis, perform the necessary preprocessing, both general and specific types appropriate for the modeling type being employed. |
| The dataset was split into two variables: the target label named ‘class’ (y) and all other columns noted as features (X). Furthermore, the two variables were placed into training and testing sets using the ‘train\_test\_split’ function (80% train, 20% test) and ‘random\_state’ equal to a fixed integer (42). Since many of the features in the dataset are frequency counts and capitalization measures, feature scaling was integrated using the ‘StandardScaler’ function from Scikit-Learn. This step is key, as standardization ensures that each feature is applied evenly to the model by transforming them into distributions with unit variance and a mean of zero. Performing this preprocessing step before model fitting for logistic regression helps to reduce the likelihood of poor performance that can often result in overfitting. |
| **Model Fitting**: explain the key steps and activities you perform to fit the model. Experiment (as appropriate) with parameters tuning. This is key, what separates highly accurate model from a less accurate ones is the amount of performance tuning performed. |
| Using the Scikit-Learn Logistic Regression function, the model was fitted using the trained scaled features (X) and the trained target label (y). Initial training was performed using default hyperparameters, which already provided strong performance. Logistic Regression was chosen for its interpretability for predicting whether an email can be labeled as spam using binary classification. Additionally, future model iterations can use ‘GridSearchCV’ from Scikit-Learn for optimization, which is often used for cross-validation to determine which hyperparameters can provide the best performance. |
| **Results** |
| **Model Properties:** explain the components of the fitted model and their characteristics. Leverage functions to summarize the model properties. Also, leverage visualization as required. |
| The fitted Logistic Regression model produced coefficients for each feature, which highlight the importance of word frequencies, character frequencies, and capitalization patterns in predicting spam. Positive coefficients show that higher feature values can increase the probability of an email being labeled as spam, while negative coefficients decrease this likelihood. For example, features such as frequent use of exclamation marks (‘char\_freq\_1’) and/or longer sequences of capital letters (‘capital\_run\_length\_average’) tend to hold stronger positive associations with spam classification. |
| **Output Interpretation**: explain the result and interpret the final model output using terms that reflect the application area and in relation to the stated objective. This is where you check whether or not the stated objective is met. |
| The model was evaluated on the test set, and the predictions were compared against the actual labels. The classification report and confusion matrix show that the logistic regression model was able to separate spam from non-spam with high accuracy. The confusion matrix revealed that the model correctly classified the majority of both spam and non-spam emails, with relatively few false positives and false negatives. These results meet the stated objective, and the model successfully achieved the goal of predicting whether an email is spam based on text-related features. |
| **Evaluation**: employ appropriate metrics to quantitatively evaluate the performance of the fitted model. For supervised classification, this includes simple accuracy, precision & recall (or sensitivity & specificity), all of which can be generated from a confusion matrix, or ROC. |
| The model achieved an accuracy of 92% (rounded to two decimal places), highlighting a reliable, strong prediction for classifying spam emails. I evaluated my model using a classification report and confusion matrix, which includes values such as precision, recall, f1-score, and support count, shown in the output below:    Results for each binary classification are noted in the classification report, along with macro and weighted averages. The F1-score’s accuracy was 0.92 and held minor variance between non-spam (0) and spam (1) values. Recall had a large difference in results, with non-spam at 95% (0) and spam at 87%. Precision held similar values to the F1-score, showing high stability within the Logistic Regression model’s accuracy of 92%. The support count highlights the frequency of spam and non-spam events in the algorithm, and its high value of 921 shows that the model can effectively identify patterns within the data.  **Confusion Matrix Heatmap and Results**  The Confusion Matrix heatmap below highlights the effectiveness of the Logistic Regression model’s performance on the Spambase dataset. The Confusion Matrix included the following results: 506 true negatives (non-spam emails), 341 true positives (spam emails), 25 false positives (non-spam emails were misclassified as spam), and 49 false negatives (spam emails were misclassified as non-spam). This information shows that the model had an overall strong performance with minor errors and a high level of accuracy. The relatively low number of false positives is important as misclassifying non-spam emails as spam can disrupt communication.    **ROC Curve Visualization (Additional Section)**  The ROC curve below further visualizes the trade-off between the true positive rate (sensitivity) and the false positive rate, holding an AUC score of 0.971 with the Logistic Regression model. The AUC score is close to 1.0, highlighting that the model is very effective at identifying whether an email is spam or non-spam. This visualization adds to the previous confusion matrix heatmap by showing that the model is well-equipped across a range of evaluation metrics. |
| **Conclusion** |
| **Summary**: highlight the main findings in relation to the stated objective. You don’t need to discuss the details of the analysis and the model such as accuracy here, just focus on the key findings. |
| This project showed that Logistic Regression is an effective supervised learning model in Machine Learning for classifying emails as spam or non-spam using UC Irvine’s Spambase dataset. The analysis highlighted that features such as word frequencies, character usage, and capitalization patterns can strongly influence the likelihood of an email being spam. The model was able to meet the stated objectives by achieving a high accuracy score of 92% and low misclassification results in the confusion matrix. The ROC curve visualization takes this a step further, holding a stable AUC score of 0.971. Overall, these key findings and results confirm that Logistic Regression can be used as a reliable algorithm for email spam detection on online communication platforms. |
| **Limitations & Improvement areas**: discuss the limitations of the analysis and identify potential improvement areas for future work. This could be related to the data, algorithm, or a combination of the two. |
| While the Logistic Regression model did have strong performance results, it is limited by its fundamental theory of linear relationships between features (X) and the predictive label (y). This may not fully capture complex patterns in larger quantities of data and can become too complex for the model, which could result in overfitting due to poor generalization. One way to improve this could include the usage of ‘GridSearchCV’ to optimize hyperparameters and further enhance the model’s overall performance. Additionally, for the most accurate and best results, more recent user data within the last 5 to 7 years should be used. UC Irvine’s dataset was collected in 1999, which can be outdated and misrepresent current user data results. |

|  |
| --- |
| **Appendix** |
|  |

**References**

Geeks for Geeks. (2025-a). *‘Advantages and Disadvantages of Logistic Regression.’* <https://www.geeksforgeeks.org/data-science/advantages-and-disadvantages-of-logistic-regression/>

Geeks for Geeks. (2025-b). *‘How to Optimize Logistic Regression Performance.’* <https://www.geeksforgeeks.org/machine-learning/how-to-optimize-logistic-regression-performance/>

University of California, Irvine. (1999). UC Irvine Machine Learning Repository. Spambase. *‘Classifying Email as Spam or Non-Spam.’* <https://archive.ics.uci.edu/dataset/94/spambase>.