|  |  |
| --- | --- |
| **DATA 440 Technical Report Assignment 1: Support Vector Machines** | **Johanson Bombaes** |
| **<enter descriptive project title here>** | |
| **URL to dataset:** | |

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. |
| Spam messages are persistent in digital communications, with billions of spam messages being sent and clogging inboxes daily. It needlessly consumes bandwidth and poses security risks by being possible vectors for malicious software or scams. The Spambase dataset from the University of Irvine (UCI) Machine Learning Repo contains a range of features extracted from emails to help determine whether a message is spam or not, with each feature representing the frequency of a specific word or character in an email.  Support Vector Machines (SVM) are suited for binary classification problems such as spam detection due to their ability to find optimal decision boundaries. By tuning kernel types and decision function shapes, we can potentially improve spam classification accuracy. |
| **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 goal of this project is to build and evaluate SVM-based models to classify emails as spam or not spam using the Spambase dataset. The primary questions we aim to answer are:   * Can SVM model accurately classify emails using the available features? * How do different configurations of SVM, passing different arguments and parameters to adjust and tune decisions functions and kernels, affect the performance? |
| **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 was sourced from OpenML via fetch\_openml(“spambase”). It consists of word frequency couts and other textual features extracted from emails. There are 57 float input features and the target variable, “class” is categorical with 1 for spam and 0 for non-spam.  By running df.isnull().sum(), we check and confirm that there were no null values. With the “class” label being category, LabelEncoder was ran to convert it into an integer. Basic exploratory steps included reviewing data structure with df.info() and distributional statistics with df.describe(). |
| **Preprocessing**: describe how you prepared the data especially for categorical inputs and categorical outputs, as well as separating data into training a testing data. |
|  |
| **Model Fitting**: explain the key steps and activities you perform to fit the model. Experiment (as appropriate) with parameters tuning: kernel and decision\_function\_shape parameters in the svm.SVC() function. This is key, what separates highly accurate model from a less accurate ones is the amount of performance tuning performed. |
|  |
| **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. |
|  |
| **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. |
|  |
| **Evaluation**: describe the metrics used to quantitatively evaluate the performance of the fitted model that are in your code: confusion matrices from the function confusion\_matrix() and other accuracy statistics from the function classification\_report(): precision, recall, f1-score and support. Interpret these statistics in the context of your model. |
|  |
| **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. |
|  |
| **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. |
|  |

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

**References**