University of Maryland Global Campus

Exercise 1b

Johanson Bombaes

DATA 440 6980 Advanced Machine Learning

Dr. Chalachew Jemberie

Dataset: Spambase

* Source : UCI Machine Learning Repo - Spambase

Spambase features are predominantly continuous values representing the frequency of specific words and characters found within each email. There are 48 features that record the percentage of words in the email that match particular keywords, such as “make,” “address,” and “free.” Additionally, six features quantify the frequency of certain characters like semicolons, parentheses, and dollar signs, which can be indicative of spam content. Furthermore, the dataset includes three features related to sequences of capital letters, including the average length of uninterrupted capital letter runs, the longest run, and the total number of capital letters present. These attributes are particularly useful for detecting stylistic patterns common in spam emails.

Statistical analysis of the dataset reveals that many of the word and character frequency features are sparse, with a large number of zeros indicating the absence of those words or characters in many emails. Conversely, the features related to capital letter usage show significant variability, with some emails exhibiting extremely long sequences of capital letters, a common tactic used in spam messages to draw attention.

The target variable in this dataset is binary, with the class label indicating whether an email is spam (1) or not spam (0). This makes Spambase a benchmark dataset for binary classification tasks and an ideal candidate for exploring a variety of machine learning algorithms such as logistic regression, support vector machines, and naive Bayes classifiers.

The article “Guide to Encoding Categorical Features Using Scikit-Learn For Machine Learning” by Jason Chong emphasizes the importance of encoding categorical data into numerical format for machine learning models, which generally cannot process text data directly. It explains the difference between nominal variables and ordinal variables like education level.

Using a student exam performance dataset, the Chong demonstrates how to encode nominal variables with Scikit-learn’s OneHotEncoder and ordinal variables with OrdinalEncoder, highlighting why these methods are preferable to Pandas’ get\_dummies and map methods. Key benefits of Scikit-learn’s encoders include seamless integration into machine learning pipelines, better management of training and test data transformations, and compatibility with hyperparameter tuning tools like GridSearch.

The article also shows how to combine both encoders using a column transformer and build machine learning pipelines incorporating these preprocessing steps along with regression models (linear regression and gradient boosting). The results indicate linear regression slightly outperforms gradient boosting in predicting students’ test scores.

Since all features besides the target column were ints or floats, I did not have to use one-hot encoding but Label Encoding was used to convert the target column from categories to int. Using LabelEncoder, the categories are converted to integers 0 (Not Spam) and 1 (Spam). This makes it suitabel for machine learning models that requirer numerical target variables like logistic regression.Graphical user interface, application

AI-generated content may be incorrect.

References:

Chong, J. (2025, January 24). Guide to encoding categorical features using Scikit-learn for machine learning. Towards Data Science. https://towardsdatascience.com/guide-to-encoding-categorical-features-using-scikit-learn-for-machine-learning-5048997a5c79/