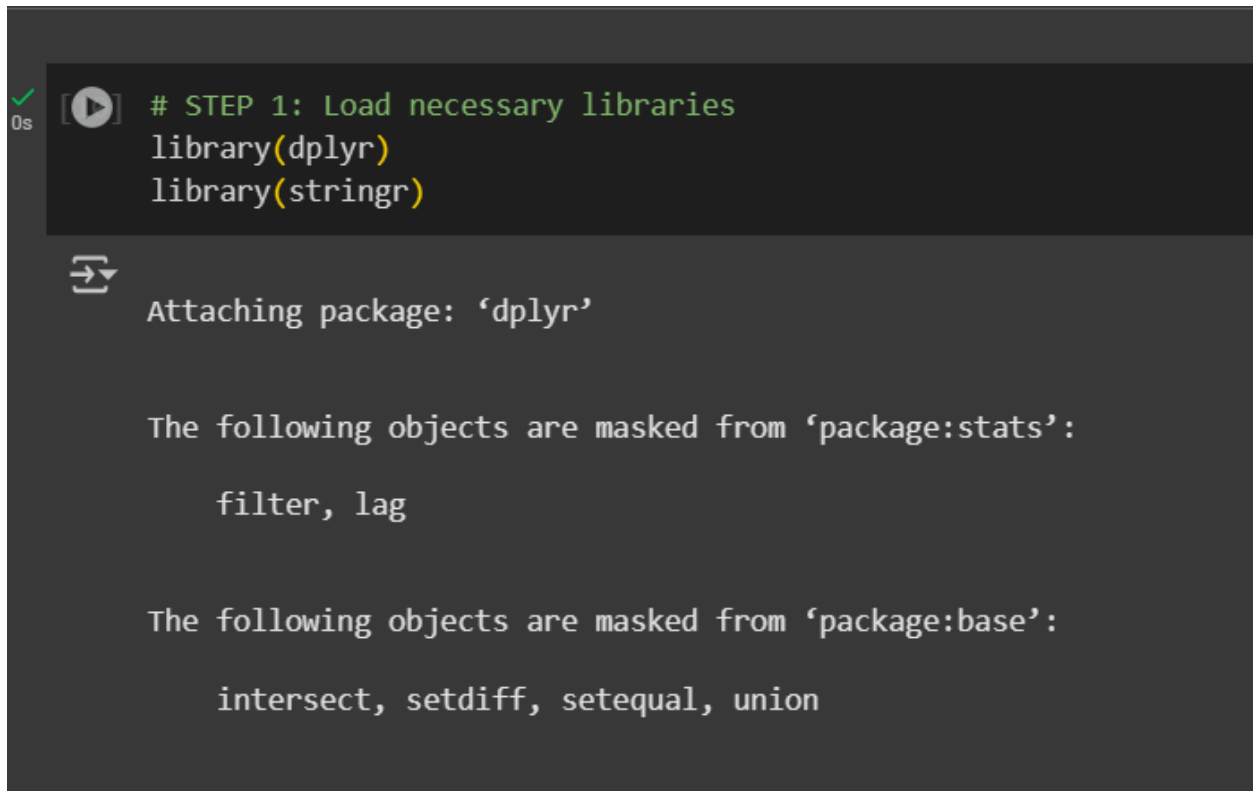


Data Labeling and Preprocessing Report: YouTube Spam Dataset

1. Objective

The objective of this project is to create a self-contained dataset of YouTube comments, accurately label each comment as spam or ham, and prepare the dataset for machine learning applications. The goal is to ensure a clean, standardized, and reproducible dataset suitable for classification tasks, including both traditional machine learning algorithms and deep learning models.

Labeling My Own Data



```
✓ 0s [▶] # STEP 1: Load necessary libraries
      library(dplyr)
      library(stringr)
```

↔

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union



STEP 2: Create my own sample dataset

```
data <- data.frame(  
  text = c(  
    "Check out my channel for free gifts!",  
    "Thanks for the tutorial, very helpful!",  
    "Click this link to win an iPhone now!",  
    "I loved this video, please make more!",  
    "Subscribe to get free money",  
    "Amazing content, learned a lot",  
    "Earn $1000 per day easily, visit here",  
    "Great explanation, thanks!"  
  ),  
  label = c(  
    "spam", "ham", "spam", "ham", "spam", "ham", "spam", "ham"  
  ),  
  stringsAsFactors = FALSE  
)
```

[13] # STEP 3: Standardize labels (lowercase, trim)

```
data <- data %>%  
  mutate(label = str_trim(str_to_lower(label)))
```

```
[13] # STEP 3: Standardize labels (lowercase, trim)
data <- data %>%
  mutate(label = str_trim(str_to_lower(label)))
```

```
[14] # STEP 4: Convert to factor
data$label <- factor(data$label, levels = c("ham", "spam"))
```

```
[15] # STEP 5: Numeric encoding (ham=0, spam=1)
data <- data %>%
  mutate(label_num = ifelse(label == "spam", 1, 0))
```

```
[16] # STEP 6: One-hot encoding for ML models
data <- data %>%
  mutate(
    ham = ifelse(label == "ham", 1, 0),
    spam = ifelse(label == "spam", 1, 0)
  )
```

```
[17] # STEP 7: Add label confidence (default = 1.0)
data <- data %>%
  mutate(label_confidence = 1.0)
```

```
# STEP 8: Quick check
print(table(data$label))
head(data)
```



```
ham spam
4    4
```

A data.frame: 6 × 6

	text	label	label_num	ham	spam	label_confidence
	<chr>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
1	Check out my channel for free gifts!	spam	1	0	1	1
2	Thanks for the tutorial, very helpful!	ham	0	1	0	1
3	Click this link to win an iPhone now!	spam	1	0	1	1
4	I loved this video, please make more!	ham	0	1	0	1
5	Subscribe to get free money	spam	1	0	1	1
6	Amazing content, learned a lot	ham	0	1	0	1



```
# STEP 9: Optional: preview text + labels
data %>%
  select(text, label, label_num, ham, spam, label_confidence)
```



A data.frame: 8 × 6

	text	label	label_num	ham	spam	label_confidence
	<chr>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
	Check out my channel for free gifts!	spam	1	0	1	1
	Thanks for the tutorial, very helpful!	ham	0	1	0	1
	Click this link to win an iPhone now!	spam	1	0	1	1
	I loved this video, please make more!	ham	0	1	0	1
	Subscribe to get free money	spam	1	0	1	1
	Amazing content, learned a lot	ham	0	1	0	1
	Earn \$1000 per day easily, visit here	spam	1	0	1	1
	Great explanation, thanks!	ham	0	1	0	1

2. Dataset Creation

A small dataset was manually constructed containing eight sample YouTube comments. Each comment was associated with an initial label:

- **Ham:** Legitimate, non-promotional comments
- **Spam:** Promotional, malicious, or unsolicited comments

The dataset was designed to simulate real-world YouTube comment scenarios, providing both positive (ham) and negative (spam) examples for classification.

3. Label Standardization

To ensure consistency:

- Labels were **converted to lowercase**.
- Leading and trailing whitespace was **trimmed**.

This step eliminates common errors from inconsistent labeling, such as variations in capitalization or extra spaces, ensuring accurate factor conversion and downstream processing.

4. Factor Conversion

The standardized labels were converted into a **factor variable** with defined levels (ham, spam). This ensures:

- Compatibility with R-based machine learning algorithms.
- Consistent reference for categorical operations.

Factor variables are essential in R to distinguish categorical data from numerical values and to support classification workflows.

5. Numeric Encoding

A numeric representation of labels was created:

- **Ham = 0**
- **Spam = 1**

Numeric encoding is critical for most machine learning algorithms that require numeric target variables. This representation facilitates logistic regression, decision trees, and neural network training.

6. One-Hot Encoding

One-hot encoding was added to represent labels in a format suitable for deep learning models:

- ham column = 1 if the comment is ham, 0 otherwise
- spam column = 1 if the comment is spam, 0 otherwise

This representation allows neural networks to predict probabilities for each class simultaneously and is standard in multi-class and binary classification tasks with categorical outputs.

7. Label Confidence

A label confidence score was introduced, with a default value of **1.0** for all examples.

- This score reflects the certainty of the label assignment.
- While initially set to full confidence, this column provides flexibility for future enhancements, such as semi-supervised learning or crowdsourced label verification.

8. Verification and Quality Checks

The dataset was validated by:

- Inspecting the **distribution of labels** to ensure a balanced representation.
- Reviewing **sample comments** and their corresponding labels to detect inconsistencies.

These checks are essential to prevent mislabeling, which could compromise model performance.

9. Dataset Structure

The final dataset includes the following fields:

1. **Text:** The YouTube comment content.

2. **Label:** Factor variable (ham/spam) for categorical classification.
3. **Label_Num:** Numeric representation of the label (0/1).
4. **Ham & Spam:** One-hot encoded columns for machine learning models.
5. **Label_Confidence:** Float value representing label reliability (default = 1.0).

This structure ensures the dataset is **machine learning-ready**, reproducible, and compatible with multiple modeling frameworks.

10. Conclusion

The data labeling pipeline provides a **robust, professional approach** for preparing a small YouTube comment dataset for classification tasks. Key features include:

- Standardized and consistent labels
- Numeric and one-hot encodings
- Label confidence for flexibility
- Verification steps for quality assurance

This dataset and labeling pipeline serve as a foundation for more advanced preprocessing, feature extraction, and model development, supporting both traditional ML and deep learning workflows.