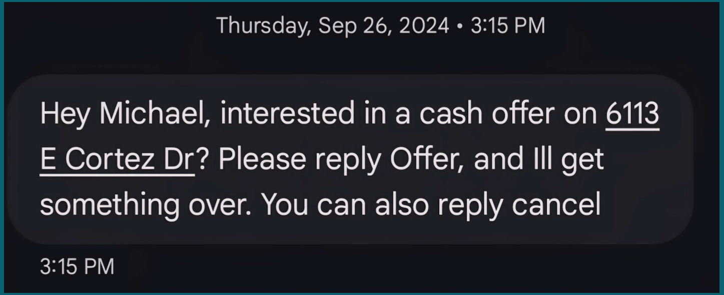
Detecting Spam Messages

1. Motivation and Approach
2. Data Acquisition
3. Analysis Plan and Challenges
4. Bias and Uncertainty
5. Results
6. Next Steps
7. References

Group 3: Taran Gupta (Group Leader),
Samantha Ritchie, Saad Shahzad

DS4002 - 001
2/16/26

Project Background



Thursday, Sep 26, 2024 • 3:15 PM

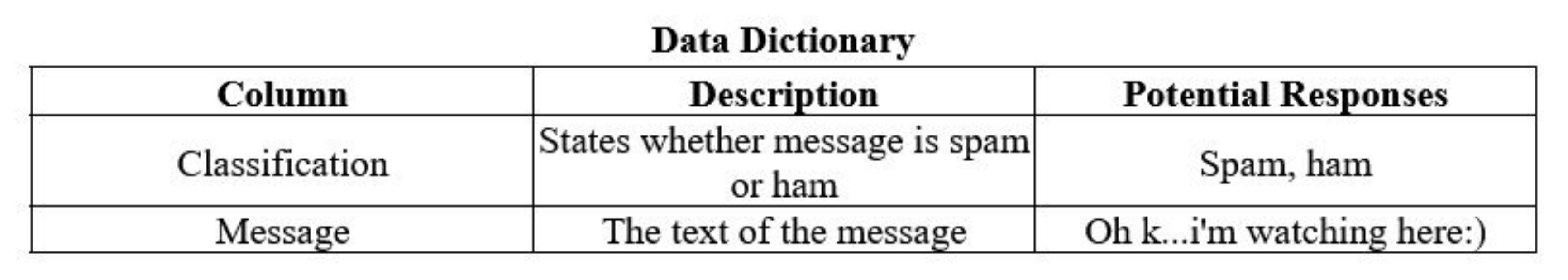
Hey Michael, interested in a cash offer on 6113 E Cortez Dr? Please reply Offer, and Ill get something over. You can also reply cancel

3:15 PM

- **Motivation**
  - Spam messages are ANNOYING and DANGEROUS
  - Understand characteristics of spam messages that differentiate them from normal messages
  - This can help people better detect spam messages, allowing them to avoid danger
- **Research question**
  - Can a classification algorithm reliably distinguish a spam message from a legitimate message based on linguistic features alone?
- **Hypothesis**
  - The addition of manually chosen variables will increase model performance when compared to a simple token vectorizer.
- **Modeling approach**
  - Naive Bayes classifier
- **Goal**
  - We will build a model that classifies SMS messages as “spam” or “ham” with at least 90 percent accuracy on a test data set.

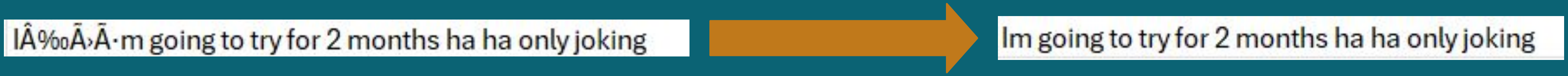
Data Acquisition

- “Text” data was acquired from Kaggle. There are no ethical or licensing concerns
- Contains 5,574 unique English messages labeled as either spam or not spam (ham)
- Data was cleaned to change any confusing symbols
- Resulting dataset was saved in csv format



Data Dictionary

Column	Description	Potential Responses
Classification	States whether message is spam or ham	Spam, ham
Message	The text of the message	Oh k...i'm watching here:)



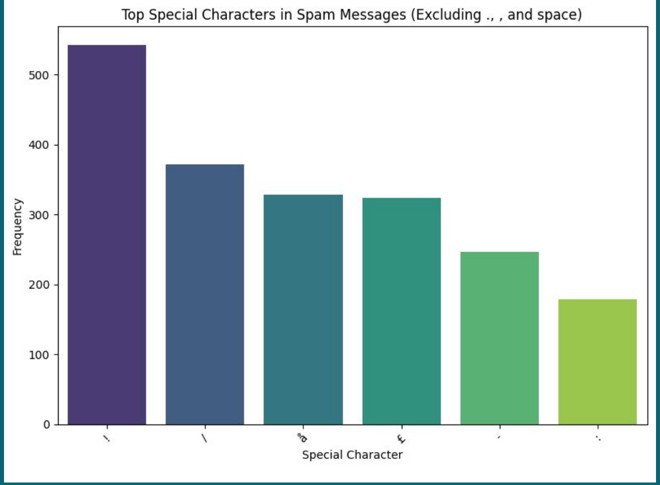
IÂ‰Ā›Ā·m going to try for 2 months ha ha only joking

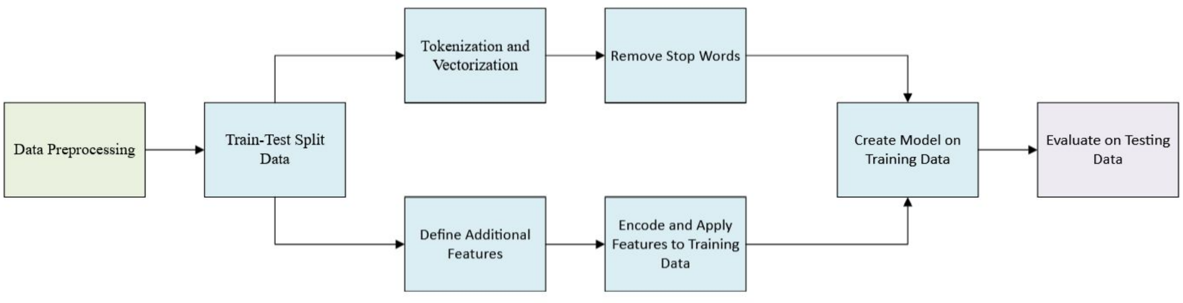
Im going to try for 2 months ha ha only joking

Analysis plan

- EDA Questions:
  - What specific words appear most frequently in spam messages ?
  - Are spam messages similar in length?
  - What special characters appear most frequently in spam messages?
- While challenging, a list of additional features was determined
- Modeling approach leverages scikit-learn

Word/Phrase	Occurrence
call	347
your	263
free	216
now	189
or	188


Top Special Characters in Spam Messages (Excluding ., , and space)



Data Preprocessing
Train-Test Split Data
Tokenization and Vectorization
Remove Stop Words
Define Additional Features
Encode and Apply Features to Training Data
Create Model on Training Data
Evaluate on Testing Data

Tricky Analysis Decision

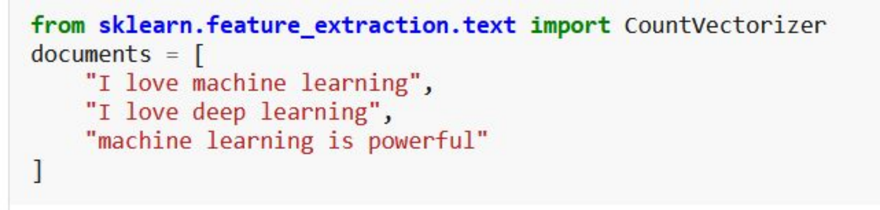
PROBLEM: Will our model (vectorizer + additional features) work well using a traditional Multinomial Distribution?

- Multinomial Distribution works best with evenly scaled inputs

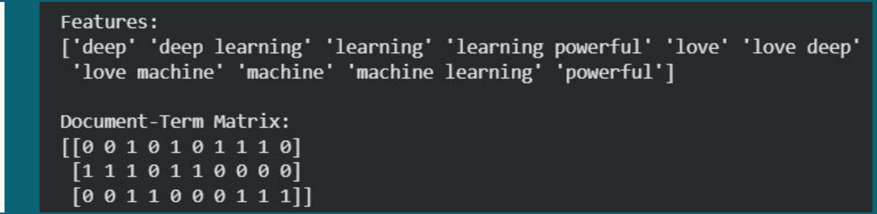
Options:

- Just use a Vectorizer (split texts into 1 token and 2 token strings)
- Combine our encoded features with a Vectorizer using a Multinomial Distribution
- Combine our encoded features with a Vectorizer using Logistic Regression
- Create ML Pipeline to determine which combination of features is most accurate

Solution: Do 3 of 4 options and compare! Slightly more work but more interesting findings



```
from sklearn.feature_extraction.text import CountVectorizer
documents = [
    "I love machine learning",
    "I love deep learning",
    "machine learning is powerful"
]
```



```
Features:
['deep' 'deep learning' 'learning' 'learning powerful' 'love' 'love deep'
 'love machine' 'machine' 'machine learning' 'powerful']

Document-Term Matrix:
[[0 0 1 0 1 0 1 1 1 0]
 [1 1 1 0 1 1 0 0 0 0]
 [0 0 1 1 0 0 0 1 1 1]]
```

Bias and Uncertainty Validation

Biases:

- Texts are from UK English → cleaned decoding errors, otherwise untouched
- Do not know who labeled the data
- ~5500 messages is a relatively small sample size

Uncertainty:

- Multiple models run
- Each model run on 3 random states (not shown)
- 80/20 Train-Test split
- Accuracy and precision measured

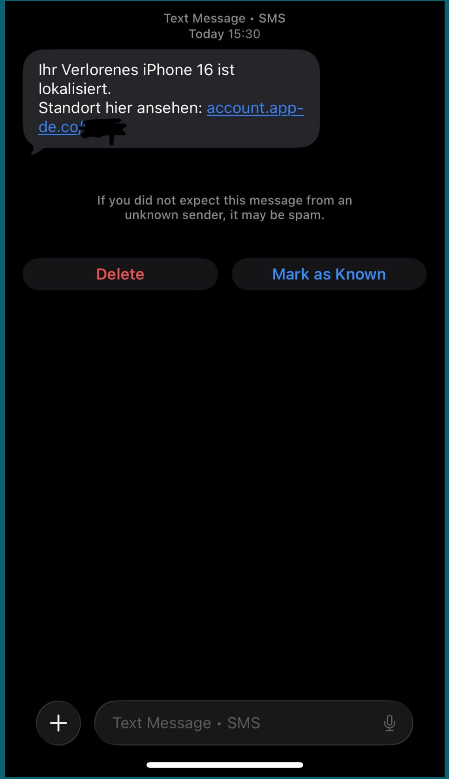
Results

Metric	Vectorizer	Features - Multinomial	Features - Regression
Accuracy	99.013%	98.386%	99.013%
Precision (Spam)	1.00	0.98	0.99
Precision (Ham)	0.95	0.98	0.98

CONCLUSION: Manually encoding additional features may not increase model performance, though logistic regression performs better than the multinomial model with this limited testing. The additional time, complexity, and performance may not be needed to achieve high accuracy detection.

Next Steps

- New lines of exploration
  - Test model on fresh datasets (different regions, different styles)
  - Incorporate sentiment analysis examining similarities in emotional tone within spam messages
- Possible improvements
  - Run on multiple random states/data for validation
  - Use ML to determine which additional features are beneficial
- New questions
  - Does message length meaningfully contribute to the classification, or are spam messages simply keyword heavy?
  - How would the model perform with messages in different languages or different linguistic styles?



Text Message • SMS
Today 15:30

Ihr Verlorenes iPhone 16 ist lokalisiert.
Standort hier ansehen: account.app-de.co/[redacted]

If you did not expect this message from an unknown sender, it may be spam.

References

Github: https://github.com/Taran-Gupta/DS4002_Project1

[1] “How to spot a scam email, text message or call,” NCSC, https://www.ncsc.gov.uk/collection/phishing-scams/spot-scams (accessed Jan. 28, 2026).

[2] Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (DocEng '11). Association for Computing Machinery, New York, NY, USA, 259–262. https://doi.org/10.1145/2034691.2034742 (accessed Jan. 28, 2026).

[3] S. Kamperis, “How to implement a naive Bayes classifier with tensorflow,” Let’s talk about science!, https://ekamperi.github.io/machine%20learning/2020/12/31/ naive-bayes-classifier-in-tensorflow.html (accessed Jan. 28, 2026).

[4] GeeksforGeeks, “Naive Bayes Classifiers,” GeeksforGeeks, Mar. 03, 2017. https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/ (accessed Feb. 04, 2026).

[5] GeeksforGeeks, “Using CountVectorizer to Extracting Features from Text,” GeeksforGeeks, Jul. 15, 2020. https://www.geeksforgeeks.org/nlp/using-countvectorizer-to-extracting-features-from-text/ (accessed Feb. 04, 2026).

[6] B. G. Regmi, “Spam Classification using Naive Bayes Algorithm,” Medium, Feb. 22, 2021. https://binitaregmi.medium.com/spam-classification-using-naive-bayes-algorithm-3e263061a3b0 (accessed Feb. 04, 2026).

[7] Ledjob, “simple_fast_classification/classification_jupyter.ipynb at master · Ledjob/simple_fast_classification,” GitHub, 2025. https://github.com/Ledjob/simple_fast_classification/blob/master/classification_jupyter.ipynb (accessed Feb. 04, 2026).

Thank you! Any questions?