Scam Detection + Prank Virus

By: Lanz Dungo, Matthew Wolz, Umair Iqbal

Spam Email Classification Using Machine Learning

Goal:

- Predict whether an email is *spam* or *not spam* using text-based features

Dataset:

- Two columns:
    1. Label (0 = not spam, 1 = spam)
    2. Text

Approach:

1. Feature Engineer text-based signals
2. Feed into a Random Forest
3. Feature Explanations

Question: *What features contribute most to whether a email is spam or not?*

Engineered Features & Model Used

Extracted:

1. Keyword indicators
   - “Free” → has_free
   - “Cash” → has_cash
   - “Link” → has_link
   - “Urgent” → has_urgent
2. Structure variables
   - Word count → num_words
   - Question marks →num_questions
3. Sentiment polarity

Trained a Random Forest Classifier (92% accuracy)

- Optuna hyperparameter tuning

Spam Label	Number of Words	Contains "Free"
0	20	0
0	6	0
1	28	1
0	11	0
0	13	0
1	32	0
0	16	0
0	26	0
1	26	0
1	29	1

Note: the full dataset has 21 columns

What Predicts Spam?

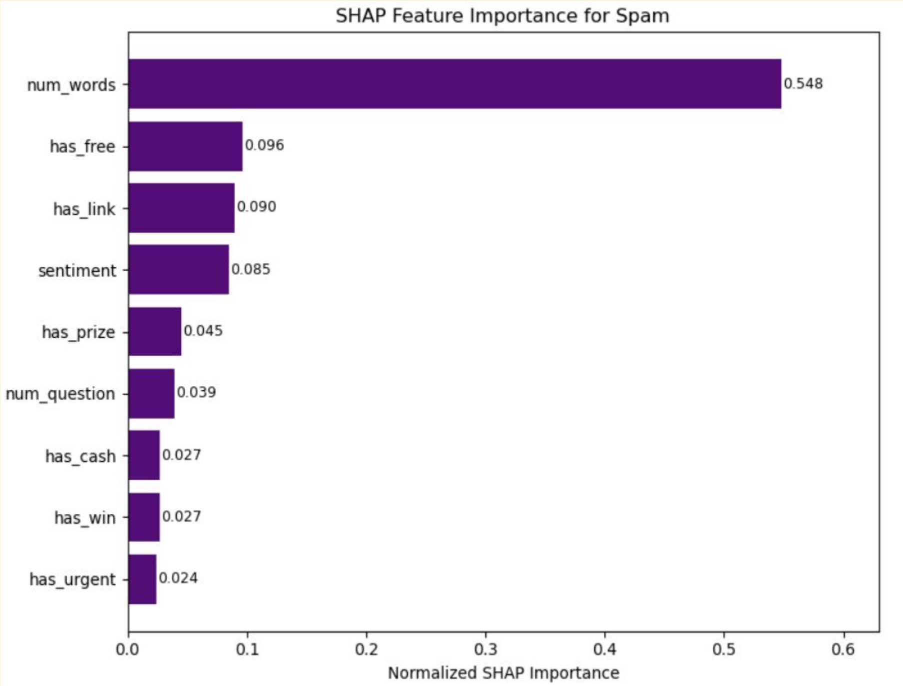
SHAP values:

- Feature ranking
- Importance values

What variables does the model deem as important?

- num_words (~55%)
- has_free
- has_link
- sentiment

SHAP values → what **signals** do the model consider “spam-like”?



How Features Affect Spam Probability

Partial Dependence Plots (PDPs)

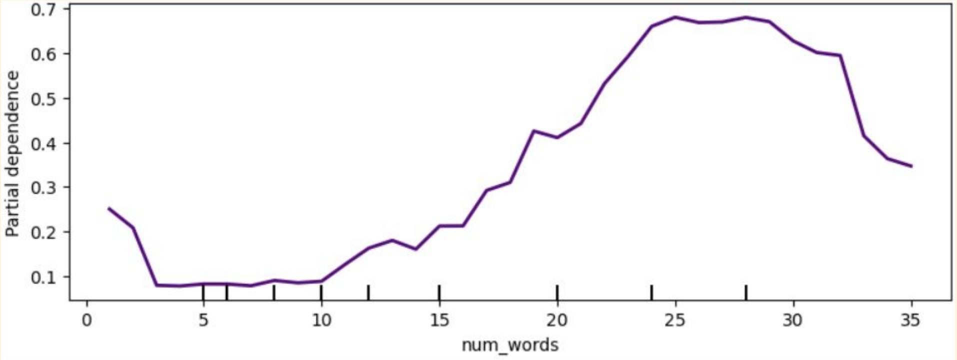
- Show how changing a single feature impacts spam probability

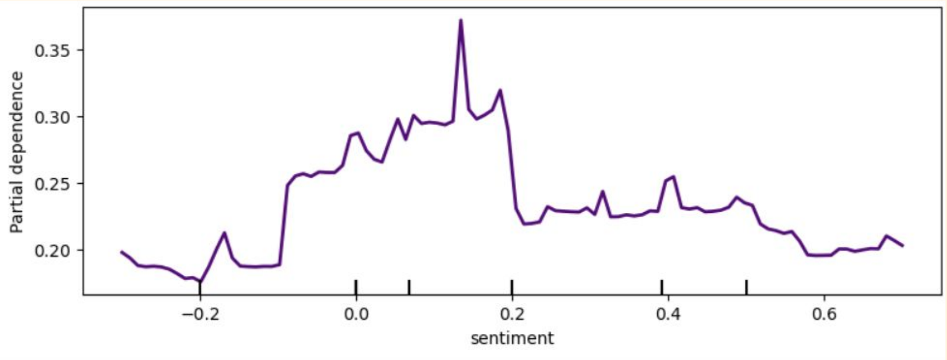
num_words

- Spam probability rises sharply after ~15-20 words in length

Sentiment

- Neutral/slightly positive sentiment increases spam likelihood





Key Takeaways

Spam emails often

- Are longer
- Contain promotional keywords (“free”, “link”, “prize”, “winner”)
- Use upbeat/positive sentiment

The **random forest** model achieves strong performance

- 92% overall accuracy
- Explainable by SHAP values + PDPs

Leads to better knowledge of what a spam email **contains**