YOUTUBE SPAM DETECTION

Artificial Intelligence for Cybersecurity

Antonio Osele

TABLE OF CONTENTS

YOUTUBE SPAM

YouTube comments are known for having lots of spam, ranging from self advertisement or irrelevant messages to straight up phishing and scam attempts. The goal of the project is to train a model able to detect such comments.

ABOUT THE DATASET

The dataset[1] contained 1956 instances of real comments extracted from five of the most viewed videos on YouTube. Each instance was labeled as spam or ham. Other attributes are: comment ID, author, date.

[1] https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection

DATA CLEANING

IMPORT

Import and concatenate the datasets

CLEAN

Remove unnecessary features

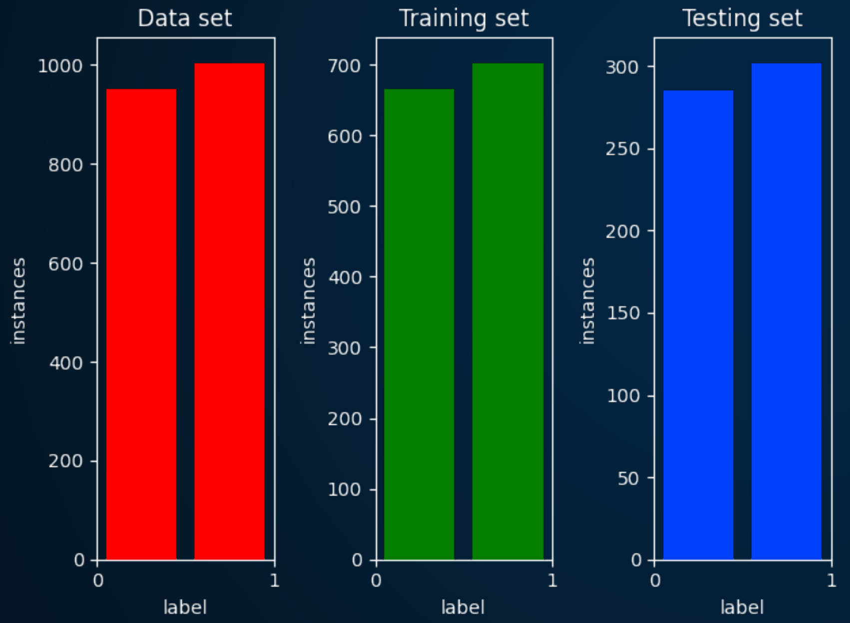
PREPROCESS

Add more useful features

DATA ANALYSIS

- HISTOGRAMS
- HEATMAP
- WORD CLOUD

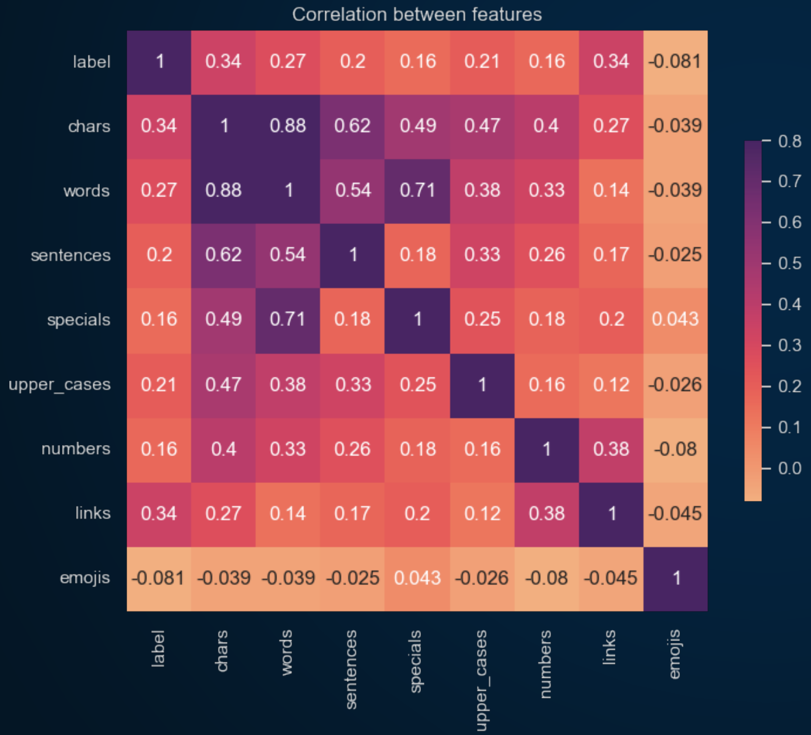
DATASET DISTRIBUTION



Balanced data

The raw dataset was already balanced. After the 70/30 split of training and testing data, the ratio between spam and ham is unchanged.

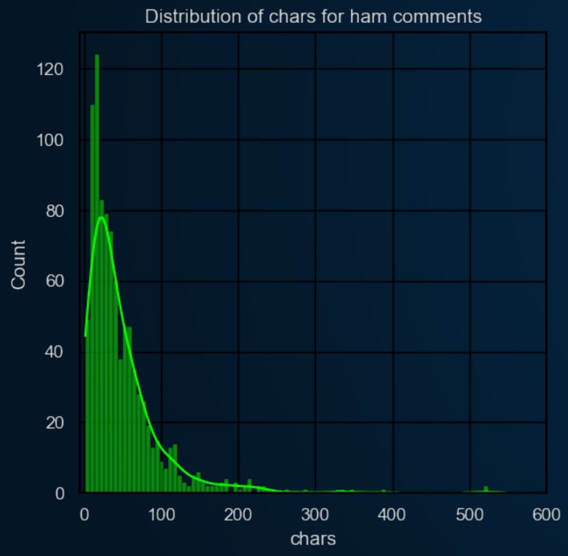
FEATURE CORRELATION

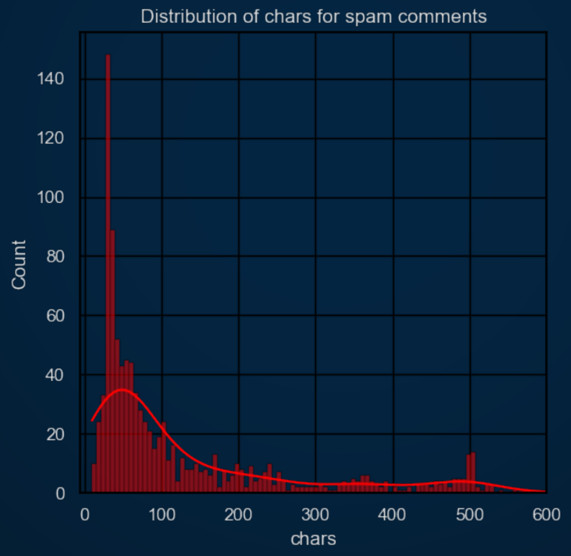


Links and emojis

Character count and links are more prevalent in spam, whereas emojis are slightly less present.

FEATURE DISTRIBUTION

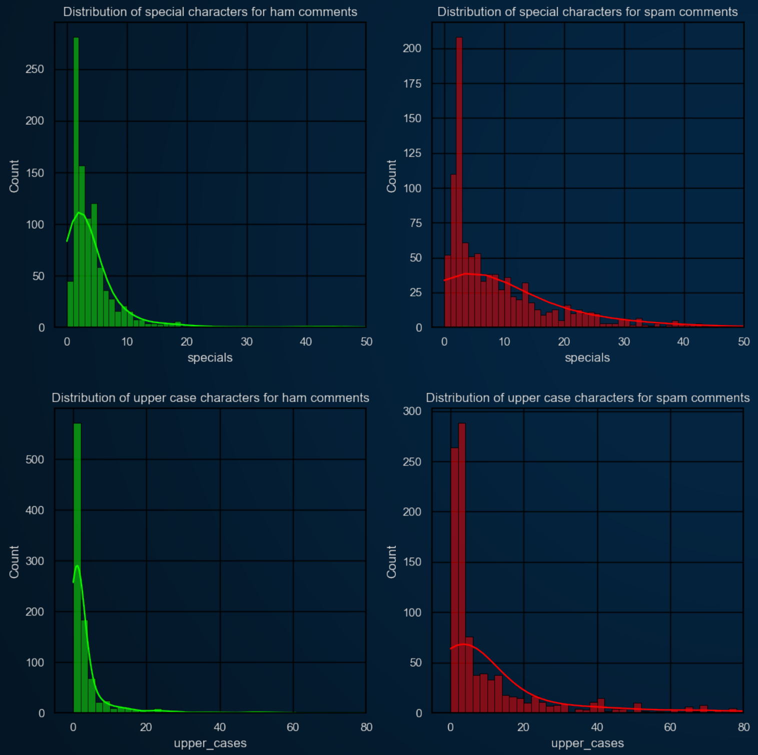




Long comments

Ham comments are on average 200 characters or less. Spam comments instead tend to be longer, with a secondary peak at around 500 characters.

FEATURE DISTRIBUTION



Other characters

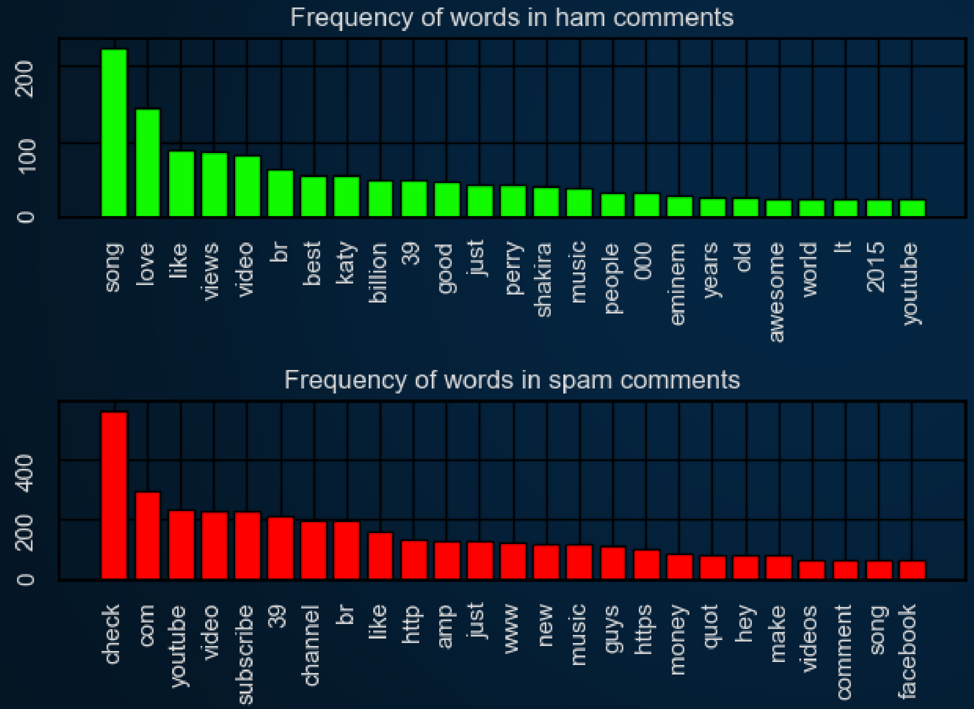
Something similar can be observed with the distribution of special and upper case characters, being more spread out in spam comments than in ham.

FEATURE DISTRIBUTION

Feature presence

Here we can see how spam are more likely to contain numbers and links. At the same time they don't have as much emojis.

WORD FREQUENCY



Common words

It's easy to see that ham comments engage normally with the video while spam comment are mostly self advertisement or phishing.

WORD CLOUD

Different view

A word cloud to show in a different way the most used words in spam comments

VECTORIZATION

Count Vectorizer was used to tokenize the text of the comments, remove accents, punctuation and stop words.

CLASSIFIERS

K-NEIGHBORS
+ simple, fast
- sensitive to outliers

GAUSSIAN WITH RBF
+ versatile (different kernels)
- inefficient if high features

SVM WITH SGD
+ fast, unbiased by outliers
- sensitive to feature scaling

SVC
+ effective in high dimensions
- sensitive to hyperparameters

CLASSIFIERS

MULTINOMIAL NB

- \+ fast, unbiased by outliers
- \- assumes all features have the same relevance

COMPLEMENT NB

- \+ same as MNB but faster on text classification tasks

DECISION TREE

- \+ easy to explain and visualize
- \- slow, prone to overfitting

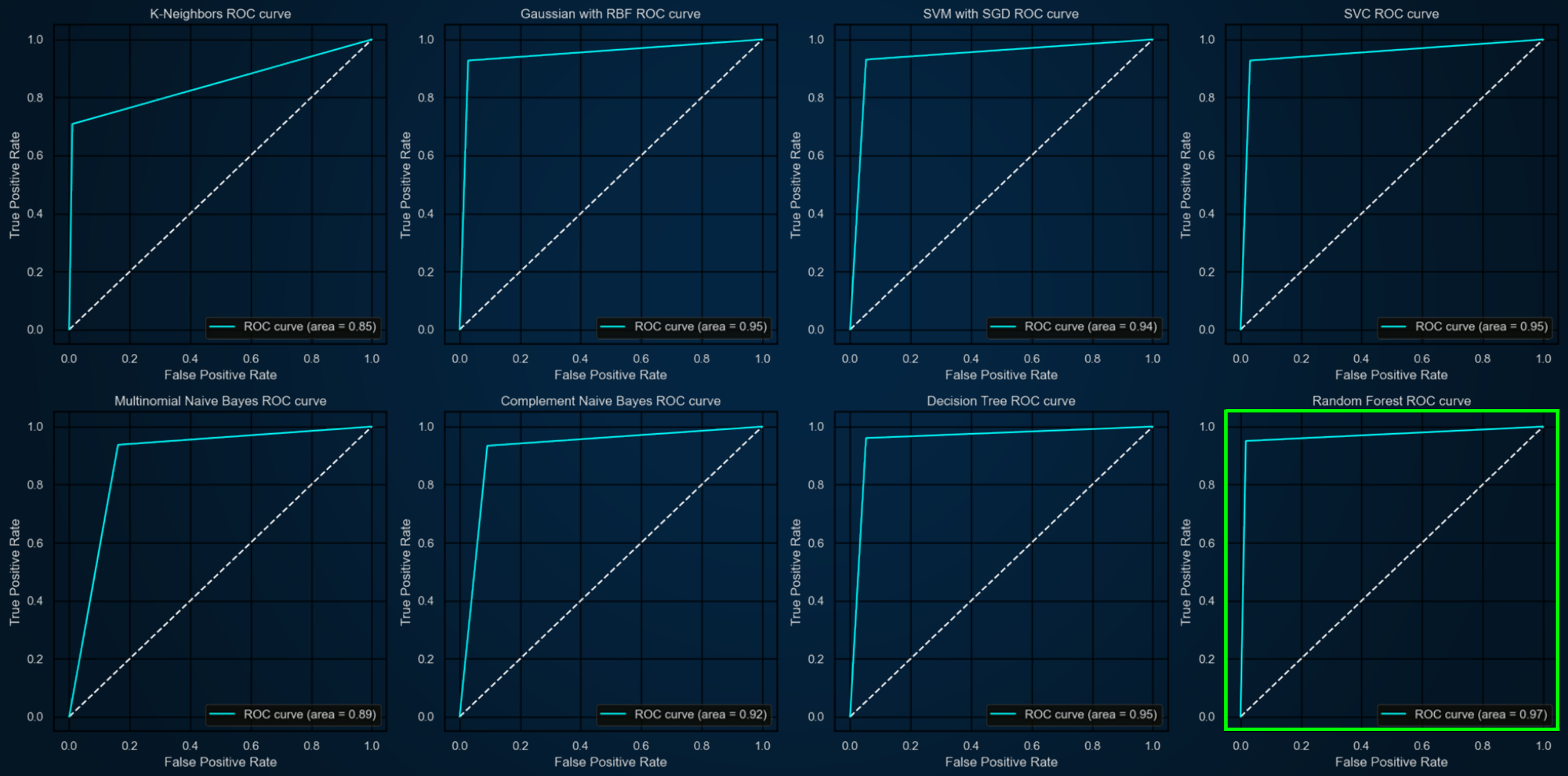
RANDOM FOREST

- \+ very accurate
- \- hard to interpret, prone to overfitting

METRICS

CLASSIFIER	CONFUSION MATRIX	ACCURACY	PRECISION	RECALL	F1
K-NEIGHBORS	[282 3] [88 214]	0.844	0.986	0.708	0.824
GAUSSIAN WITH RBF	[277 8] [22 280]	0.948	0.972	0.927	0.949
SVM WITH SGD	[270 15] [21 281]	0.938	0.949	0.930	0.939
SVC	[276 9] [22 280]	0.947	0.968	0.927	0.947
MULTINOMIAL NAIVE BAYES	[239 46] [19 283]	0.889	0.860	0.937	0.896
COMPLEMENT NAIVE BAYES	[259 26] [20 282]	0.921	0.915	0.933	0.924
DECISION TREE	[270 15] [12 290]	0.954	0.950	0.960	0.955
RANDOM FOREST	[280 5] [15 287]	0.965	0.982	0.950	0.966

ROC CURVES


K-Neighbors ROC curve
ROC curve (area = 0.85)
Gaussian with RBF ROC curve
ROC curve (area = 0.95)
SVM with SGD ROC curve
ROC curve (area = 0.94)
SVC ROC curve
ROC curve (area = 0.95)
Multinomial Naive Bayes ROC curve
ROC curve (area = 0.89)
Complement Naive Bayes ROC curve
ROC curve (area = 0.92)
Decision Tree ROC curve
ROC curve (area = 0.95)
Random Forest ROC curve
ROC curve (area = 0.97)
True Positive Rate
False Positive Rate

STRATIFIED K-FOLD CROSS-VALIDATOR

CLASSIFIER	CONFUSION MATRIX	AVG ACCURACY	AVG PRECISION	AVG RECALL	AVG F1
K-NEIGHBORS	[928 23] [308 697]	0.831	0.860	0.835	0.828
GAUSSIAN WITH RBF	[910 41] [77 928]	0.940	0.942	0.940	0.940
SVM WITH SGD	[889 62] [78 927]	0.931	0.941	0.930	0.924
SVC	[884 67] [90 915]	0.920	0.922	0.920	0.920
MULTINOMIAL NAIVE BAYES	[816 135] [97 908]	0.881	0.884	0.881	0.881
COMPLEMENT NAIVE BAYES	[836 115] [100 905]	0.890	0.892	0.890	0.890
DECISION TREE	[890 61] [72 933]	0.932	0.934	0.926	0.929
RANDOM FOREST	[891 60] [67 938]	0.935	0.938	0.936	0.930

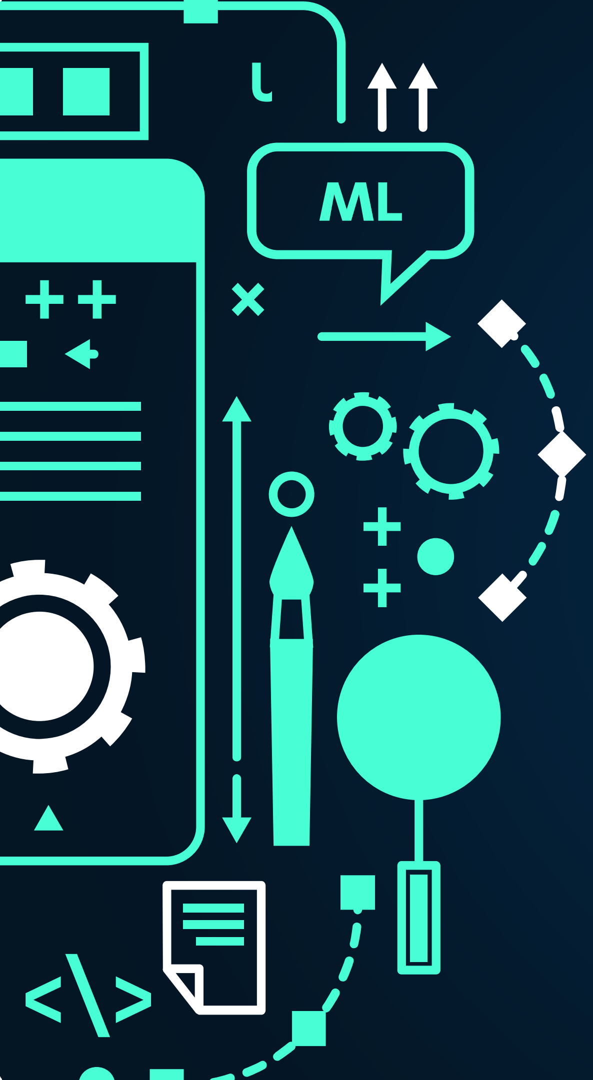
CONCLUSIONS

RESULTS

Even if the dataset wasn't very big it achieved acceptable results and the model is able to correctly classify most comments. Overall the initial goals of the project were reached.

POSSIBLE IMPROVEMENTS

- Collect more data to expand the data set
- Optimize with Grid/Random search
- Discriminate benign links
- Test with comments from other videos



THANKS!

Antonio Osele
647926
a.osele@studenti.unipi.it