Email Spam Classification

Spam is becoming even more prevalent as automated bots are becoming increasingly advanced, mimicking human language and behavior in even more convincing ways. Classification problems present a challenge in minimizing:

False Positives: a regular email classified as spam
False Negatives: a spam email classified as a regular email

In classification tasks, the question of tuning a model as to limit one or the other presents a tough choice. Additionally, one ‘class’ can be a minority of the dataset, but be very important to classify 100% correctly. For example, the classification of a life-threatening, but rare condition, those with the disease would be a minority of the data, but the cost of classifying those with the condition as healthy would be catastrophic. On the other hand, classifying those without the disease as having it, would just mean a follow-up test. The priority for such a project would be greatly minimizing **false negatives**, a diseased subject classified as healthy.

In 2023, studies have shown **46%** of emails are spam, which is relatively balanced. Additionally, the cost of a regular email being caught as spam is a lot higher than a spam email being allowed into the inbox. For spam emails, I chose to **minimize false positives**..

Generalized Training: Spam Classification

The words (that indicate spam) in this dataset will likely be different from other datasets. The goal is to evaluate which algorithms, scaling, split_method, and other methods for their strengths in training on labeled spam datasets.

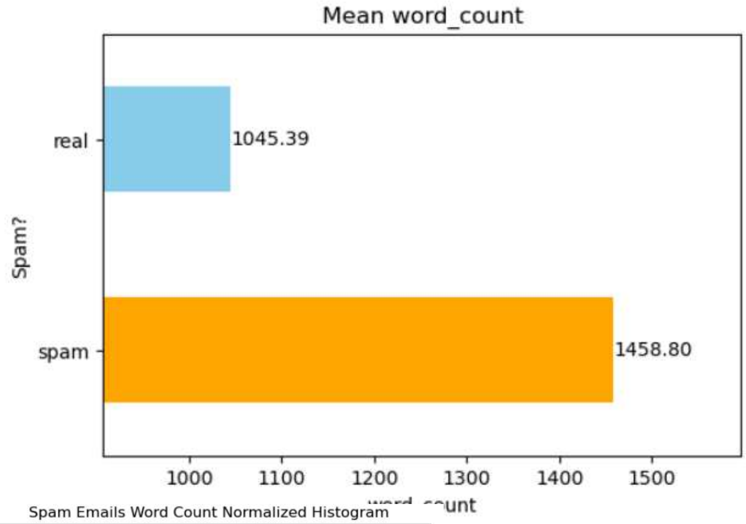
Which training methodology will be best suited to train a…

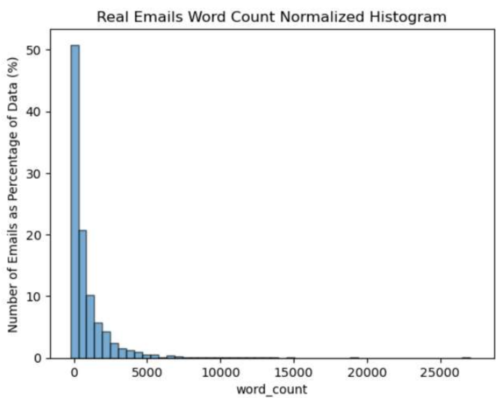
1) ‘Strict’ Spam filter: Never Misses a Spam Email
2) ‘Relaxed’ Spam filter: Never Misclassifies a Real Email as Spam
3) ‘Balanced’ Spam filter: Least Number of Errors - period

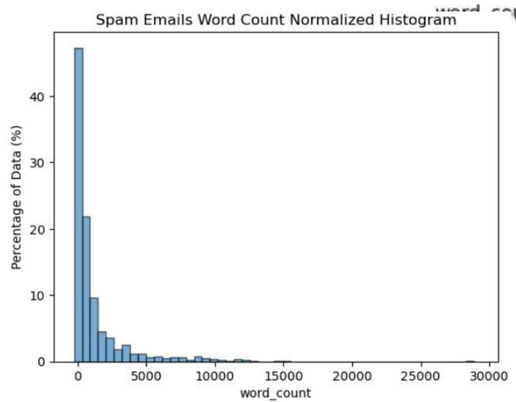
EDA

Word Counts:
Real Emails vs. Spam Emails

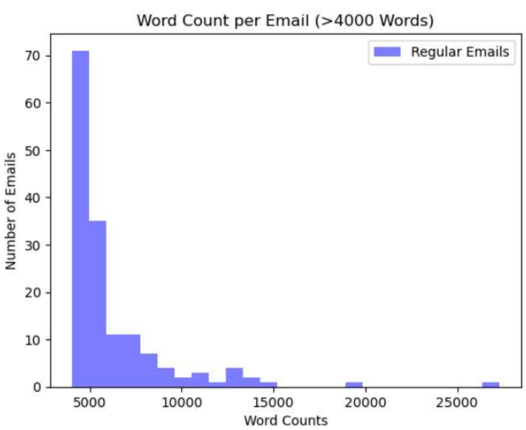
The distribution of word counts are very similar, when taking into account the class imbalance through normalization.

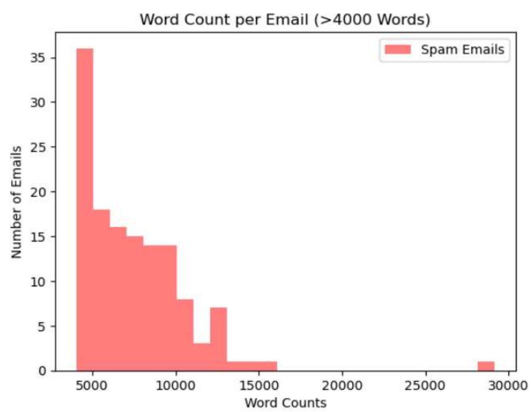






Spam contains a higher number of emails above ~6,000 words.





SPAM EMAILS

count	1500.00
mean	1458.79
std	2283.84
min	8.00
25%	316.00
50%	632.00
75%	1506.00
max	29178.00

REG EMAILS

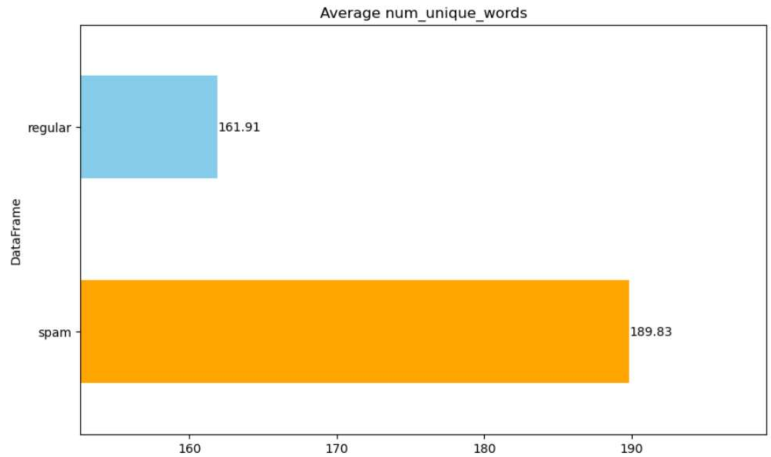
count	3672.00
mean	1045.39
std	1478.04
min	21.00
25%	244.00
50%	551.50
75%	1293.75
max	27319.00

Number of Unique Words

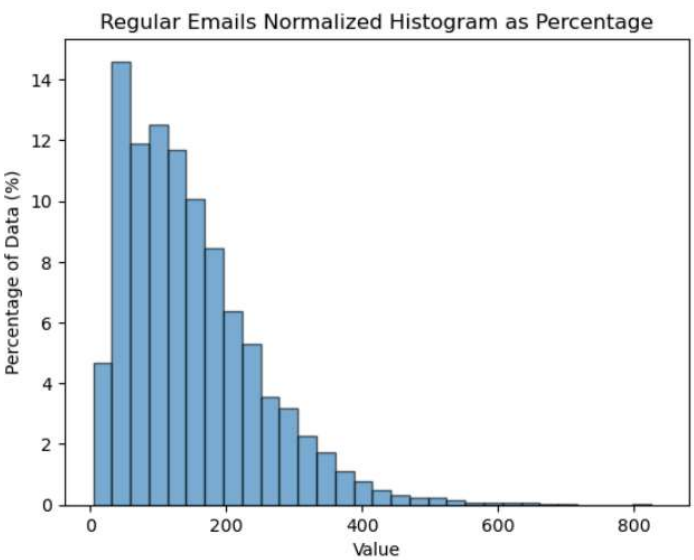
The distribution of ‘word count’ by email is fairly similar, with the number of emails.

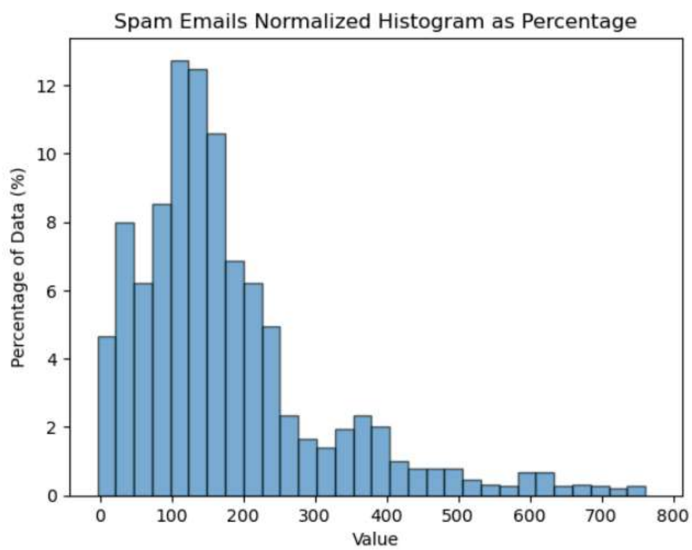
REG EMAILS	
count	3672.00
mean	161.90
std	100.07
min	18.00
25%	87.00
50%	142.00
75%	214.00
max	839.00

SPAM EMAILS	
count	1500.00
mean	189.83
std	136.47
min	9.00
25%	103.00
50%	155.00
75%	234.00
max	774.00



Different distributions





Regular emails skewed towards 25-50 unique words.
Spam emails skewed towards 150-175 unique words.
Spam emails have a higher percentage of emails above 400 unique words.

25 Least Correlated & 25 Most Correlated Words with Spam

Logistic Regression

Less Spam

Feature	Coefficient
enron	-2.076216
hpl	-1.256958
hp	-1.007039
deal	-0.907440
tax	-0.873753
attached	-0.800511
have	-0.731578
nom	-0.723420
hou	-0.720882
if	-0.719598
aren	-0.674872
meter	-0.656242
tu	-0.647107
da	-0.630307
xls	-0.603930
met	-0.595181
please	-0.593304
daren	-0.591907
mb	-0.586000
list	-0.583050
att	-0.575460
gas	-0.557245
question	-0.555821
file	-0.546511
thank	-0.543298

More Spam

Feature	Coefficient
one	1.206206
z	1.083168
mo	1.017817
rm	0.963883
men	0.939673
ali	0.838998
sa	0.837437
ur	0.804109
ca	0.800978
only	0.761055
ad	0.737519
http	0.731307
ii	0.689261
ve	0.660917
gr	0.658574
money	0.647384
gra	0.636383
of	0.630562
hi	0.622273
our	0.611170
here	0.585819
pa	0.585172
dr	0.577433
no	0.577027
her	0.563221

Point BiSerial

Less Spam

Feature	Correlation
thanks	-0.271433
hpl	-0.266518
hanks	-0.266070
thank	-0.262384
attached	-0.236558
daren	-0.236180
forwarded	-0.230765
subject	-0.227754
hp	-0.225846
aren	-0.206063
nom	-0.202600
farmer	-0.194693
questions	-0.193163
deal	-0.190407
than	-0.188514
volume	-0.188005
enron	-0.186740
question	-0.185967
xls	-0.179113
meter	-0.166499
please	-0.162304
btu	-0.162183
pm	-0.161234
mmbtu	-0.157753
gas	-0.156652

More Spam

Feature	Correlation
more	0.258152
our	0.228187
able	0.222219
best	0.221703
ur	0.220253
sex	0.220092
sec	0.217402
money	0.217215
soft	0.213382
dr	0.212413
mo	0.210056
via	0.204031
prescription	0.203896
remove	0.203384
cheap	0.200348
meds	0.198501
drug	0.197976
of	0.197234
ali	0.194936
ic	0.194706
cia	0.191742
offer	0.190368
off	0.189532
your	0.186149
prices	0.186026

Class Weights

Because of 2023 studies showing 46% spam in reality and taking into account the 71% / 29% class imbalance, class weights can be set at an inverse proportionality to mimic this split that uses roughly **double** class weight for spam. However, with spam classification we want to **increase the cost of a false positive**, in order to limit a regular email being marked as spam. Therefore, the class weights I found optimal after testing was **28% more weight for spam.** (class_weight = {0: 1, 1: 1.28}) 'mimics' a 63% / 37% split for real / spam email representation.

Modeling

- **Scalars**: MinMax, MaxAbs, Standard
- **Split Methods**: TrainTestSplit, StratifiedKFold, StratifiedShuffleSplit
- **Models**: RandomForest, CatBoost, LGBM, LogisticRegression
- **Sampling:** SMOTE vs. 'Un-SMOTE'
- **Class Weights:** Spam Weighted 128% vs. Unweighted

Results: 3 Optimized Model Classes

#1 'Relaxed' Spam Filter: No real emails in spam

- **100% real:** *Logistic Regression, SSS, MinMax/MaxAbs, SMOTE*
    - 74% Threshold: <u>13.33% of spam allowed</u>
        - **Alt**: *CatBoost, SKF, unSMOTE, unweighted*
        - 82% Threshold: 16.33% of spam allowed

- **99.5% real:** *Logistic Regression, SSS, MinMax/MaxAbs, Un-SMOTE*
    - 58.61% Threshold: <u>8.33% of spam allowed, 0.36% real emails in spam</u>
        - **Alt:** *CatBoost, SKF, SMOTE, Weighted*
        - 84% Threshold: 8.7% spam allowed, 0.36% real emails in spam

#2 'Strict' Spam Filter: No spam missed

- **100% spam:** *RandomForest, TT, MinMax, SMOTE, Unweighted*
    - 49% Threshold: <u>7.82% of real emails marked as spam</u>
        - **Alt:** *CatBoost, TrainTest, Any Scaler, Un-SMOTE, Unweighted*
        - 23% Threshold: 11.88% of real emails marked as spam

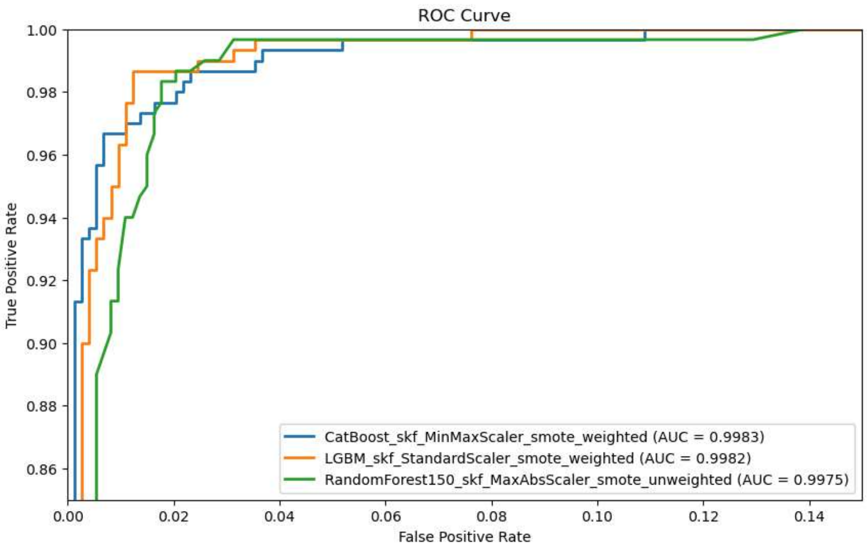
- **99.5% spam:** *RandomForest, SSS, MinMax, SMOTE, Unweighted*
    - 48% Threshold: <u>5.67% of real emails marked as spam, 0.33% of spam missed</u>
        - **Alt:** *LGBM, SKF, Any Scaler/Smote/Weights*
        - 16% Threshold: 7.72% of real emails marked as spam, 0.33% of spam missed

#3 **'Balanced' Filter:** Harmonic mean of missed spam and real into spam

- LightGBM, SKF, StandardScaler, SMOTE
    - 57% Threshold: 0.9785 f1-score
    - 2.95% of real emails in spam box, 1.33% of spam emails in inbox

- LGBM, SSS, MinMax/MaxAbs, unSMOTE
    - 52% Threshold: 0.9735 f1-score
    - 3.39% of real emails in spam box, 2% of spam emails in inbox

ROC AUC

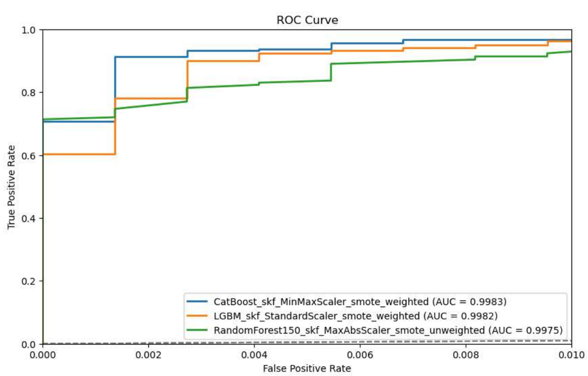


True Positive Rate (Y-axis): TP / TP + FN (Recall)
Measures the Number of True Positives for every False Negative.
'Spam Classification Rate': Number of spam classifications for every actual spam email

False Positive Rate (X-axis): FP / FP + TN
Measures the Number of False Positives for every True Negative.
'Regular Misclassification Rate': Number of regular emails caught as spam for every actual regular email

Top of the graph: the **farthest left** curve performs with the best precision at 100% recall.

LGBM, SKF, Agnostic towards: Scaler, Smote, Weights



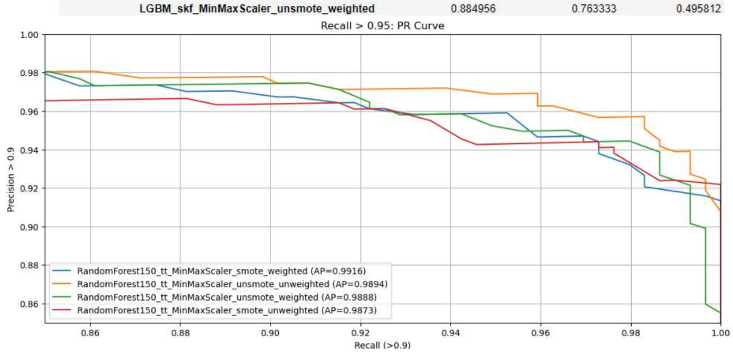
Left of the graph: the **highest** curve classifies the highest number of spam (recall) at the moment it makes its first mistake in classifying the first regular email as spam

CatBoost, SKF, Any Scaler, Smote, Weights

‘Strict’ Spam Filters (‘No Spam Allowed’)

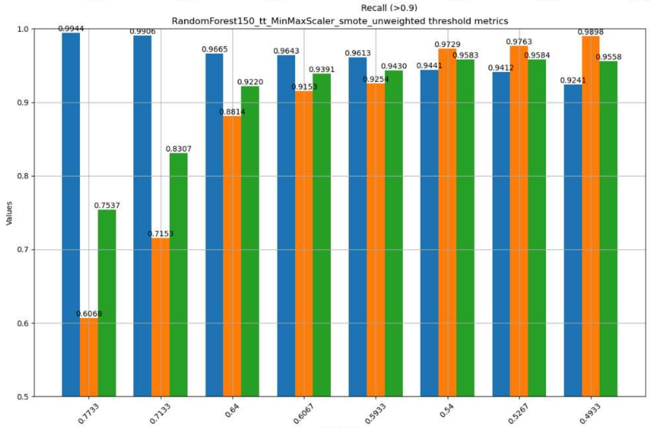
model	precision_at_recall_1	recall_at_precision_1	optimal_threshold
RandomForest150_tt_MinMaxScaler_smote_unweighted	0.921875	0.477966	0.493333
RandomForest150_tt_MinMaxScaler_smote_weighted	0.913313	0.633898	0.546667
RandomForest150_tt_MinMaxScaler_unsmote_unweighted	0.907692	0.359322	0.486667
RandomForest150_tt_MaxAbsScaler_unsmote_weighted	0.902141	0.416949	0.480000
RandomForest150_tt_MaxAbsScaler_unsmote_unweighted	0.899390	0.427119	0.513333
CatBoost_tt_StandardScaler_unsmote_unweighted	0.891239	0.508475	0.434900
RandomForest150_tt_StandardScaler_unsmote_unweighted	0.891239	0.457627	0.493333
CatBoost_tt_MinMaxScaler_unsmote_unweighted	0.891239	0.508475	0.434900
CatBoost_tt_MaxAbsScaler_unsmote_unweighted	0.891239	0.508475	0.434900
CatBoost_tt_StandardScaler_smote_weighted	0.888554	0.400000	0.709200
CatBoost_tt_MinMaxScaler_smote_weighted	0.888554	0.400000	0.709200
CatBoost_tt_MaxAbsScaler_smote_weighted	0.888554	0.400000	0.709200
LGBM_skf_MinMaxScaler_unsmote_weighted	0.884956	0.763333	0.495812

RandomForest150
MinMax/MaxAbs
TrainTest
SMOTE
Unweighted



Effect of SMOTE:
Increased Precision at 100% Recall,
Orange to Red, Green to Blue

Effect of Class Weights:
Decreased Precision at 100% Recall
Orange to Green, Red to Blue.

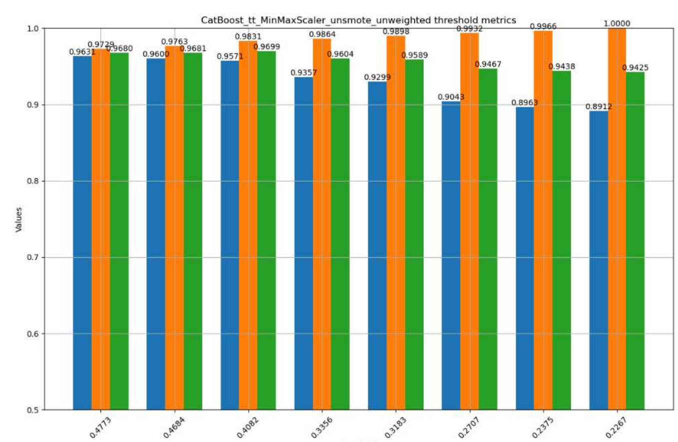


RandomForest150, MaxAbs, Train/Test, SMOTE, unweighted
49% thresh
7.82% of real emails caught as spam
F1: 0.9593

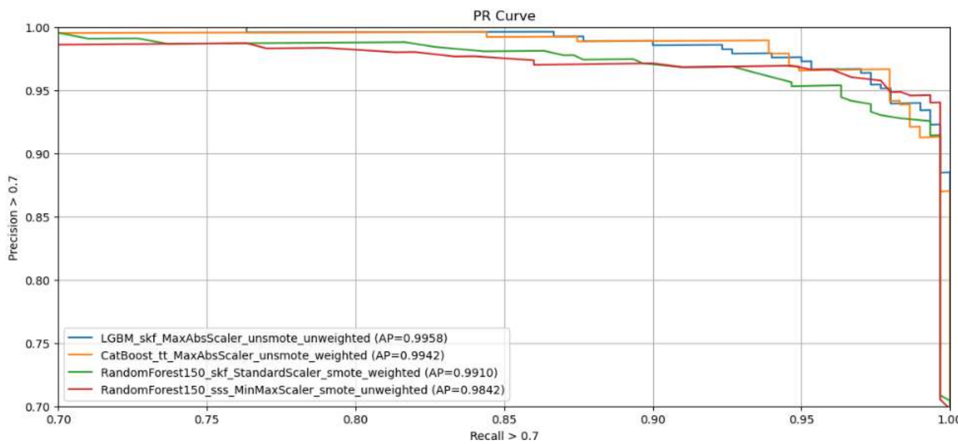
As a ‘backup’ option, CatBoost may generalize better to new data. However, it performs slightly worse on this dataset:

CatBoost, TrainTest, any scaler, Un-SMOTE, Unweighted

23% thresh
11.88% of real emails caught as spam
f1: 0.9425
(**Smote/Weighted option marginally worse)**



Over 99.5% Recall

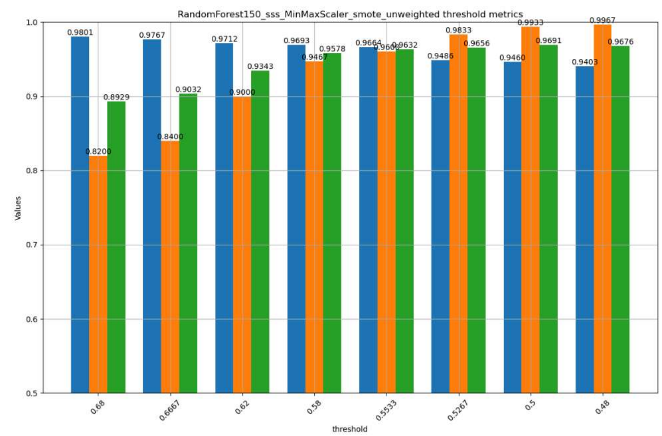


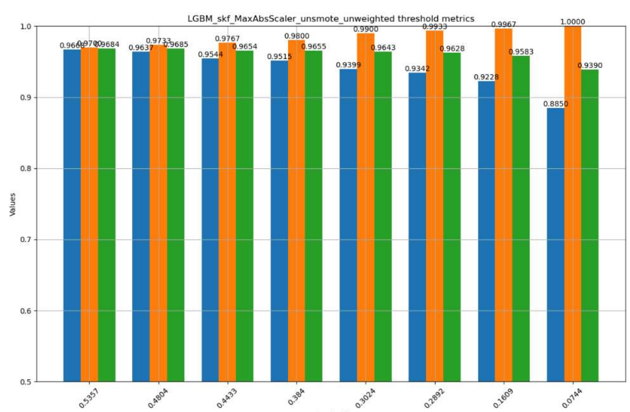
index	max_f1
RandomForest150_sss_MinMaxScaler_smote_unweighted	0.967638
RandomForest150_sss_MaxAbsScaler_smote_weighted	0.961415
LGBM_skf_MaxAbsScaler_unsmote_unweighted	0.958333
LGBM_skf_MaxAbsScaler_unsmote_weighted	0.958333
LGBM_skf_StandardScaler_unsmote_unweighted	0.958333
LGBM_skf_StandardScaler_unsmote_weighted	0.958333
LGBM_skf_MinMaxScaler_unsmote_unweighted	0.958333
LGBM_skf_MinMaxScaler_unsmote_weighted	0.958333
LGBM_skf_StandardScaler_smote_unweighted	0.956800
LGBM_skf_StandardScaler_smote_weighted	0.956800
RandomForest150_skf_StandardScaler_smote_weighted	0.953748
CatBoost_tt_MaxAbsScaler_unsmote_weighted	0.952998
CatBoost_tt_StandardScaler_unsmote_weighted	0.952998
CatBoost_tt_MinMaxScaler_unsmote_weighted	0.952998
LGBM_skf_MinMaxScaler_smote_weighted	0.952229

RF150: Steep drops when increasing threshold to classify all spam as spam (100% recall)

RandomForest150, SSS, MinMax, SMOTE, Unweighted

Threshold: **48%**
f1: **0.9676**
5.67% of real emails caught as spam
0.33% of spam missed

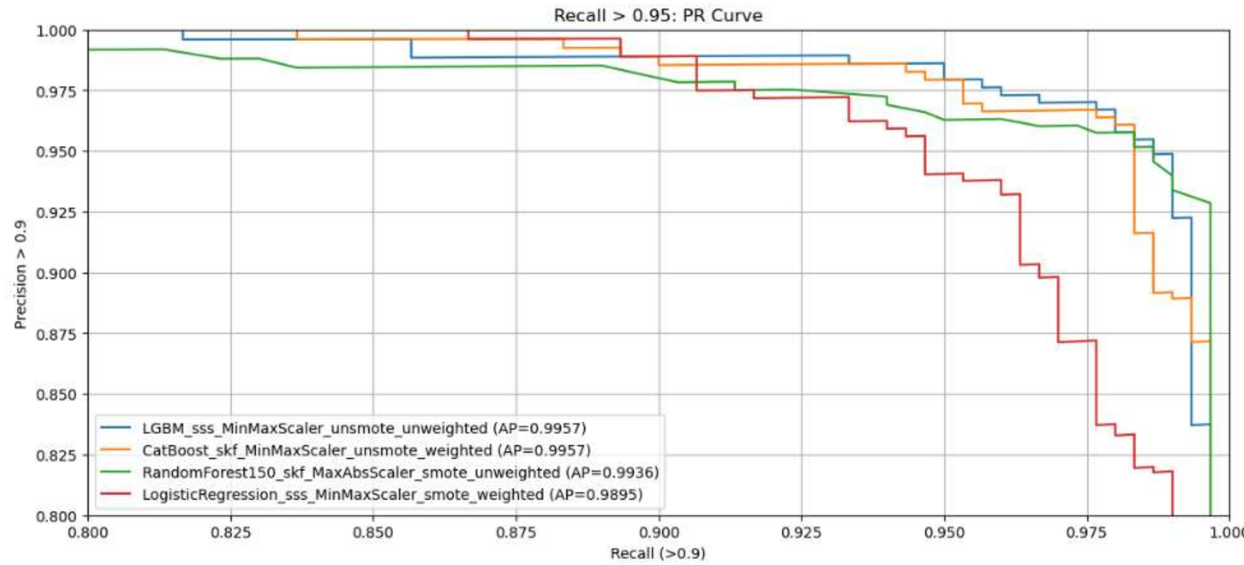




LGBM, SKF, Agnostic towards SMOTE/Weights/Scaling
Except StandardScaler w/ smote

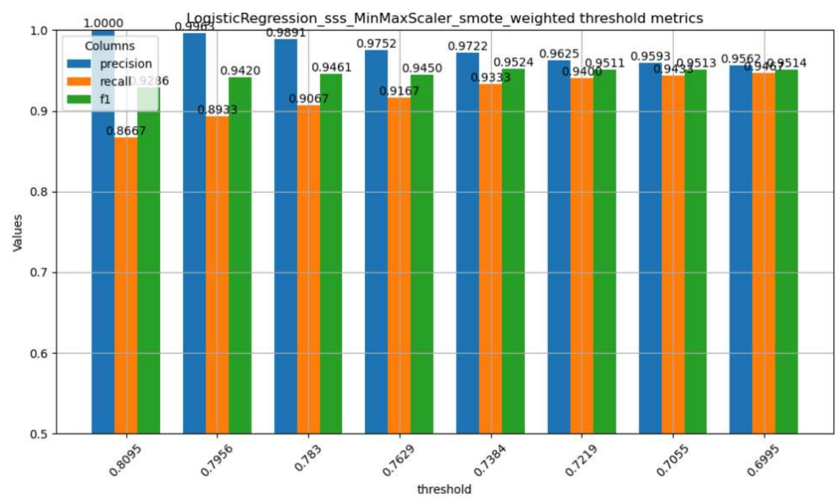
Threshold: **16%**
f1: **0.9583**
7.72% of real emails caught as spam
0.33% of spam missed

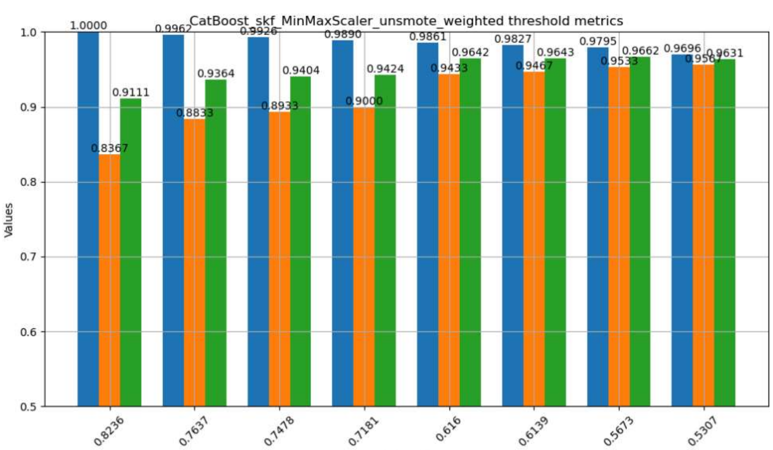
‘Relaxed’ Spam Filter: Top Recall with 100% Precision



LogisticRegression, SSS, MinMax/MaxAbs, SMOTE

13.33% Spam Allowed into inbox
f1: 0.9286



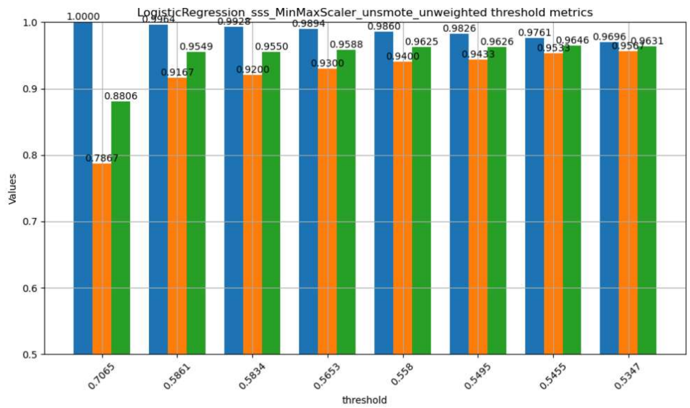


CatBoost, SKF, (any scaler), unsmote, weighted

82.36% threshold
No real emails caught as spam
16.33% spam emails into inbox
F1: 0.9111

Greater than 99.5% Precision

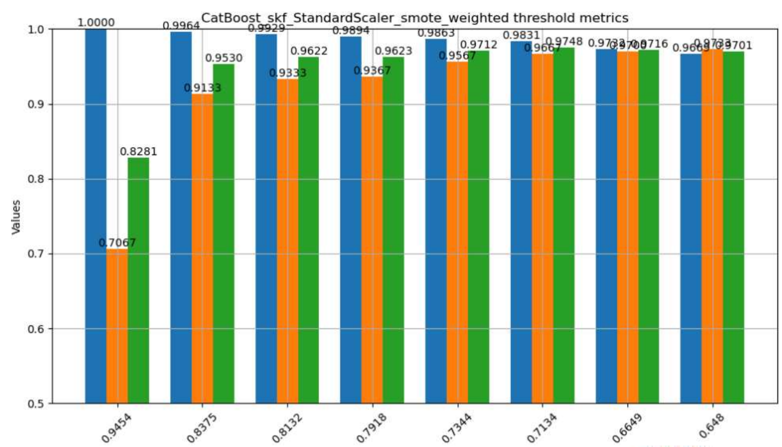
LogisticRegression, SSS, MinMax/MaxAbs, UnSmote, Unweighted/Weighted



Threshold: 58.61%
0.36% Real emails caught as spam
8.33% spam emails in inbox
f1: 0.9549

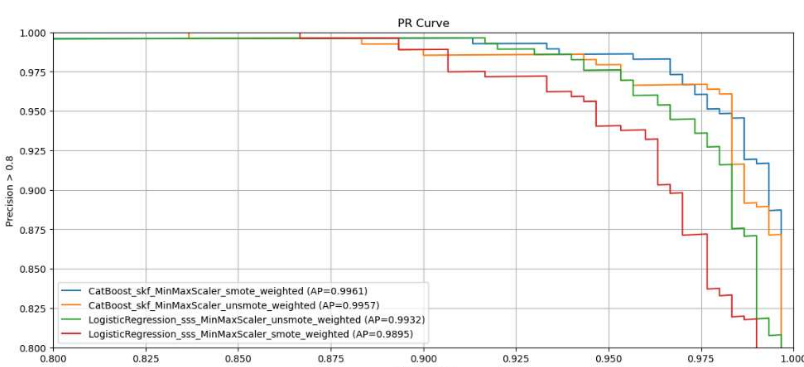
CatBoost, SKF, SMOTE, weighted
Threshold: 84%
0.36% Real emails caught as spam
8.70% spam emails in inbox
f1: 0.9530

Only loses 0.0034 recall, performs with higher f1 at lower precisions



Despite LR performing best at >99.5% Precision, it quickly falters, and would likely not be a viable long-term solution. The average_precision, f1, roc_auc all show weaknesses. Not far behind in performance at 100% precision is CatBoost, which is more likely to provide a valuable spam detection with zero false positives, despite having slightly worse recall at higher precision.

index	max_f1
LogisticRegression_sss_MinMaxScaler_unsmote_weighted	0.954861
CatBoost_skf_MinMaxScaler_smote_weighted	0.953043
LogisticRegression_sss_MinMaxScaler_smote_weighted	0.942004
LogisticRegression_sss_StandardScaler_smote_weighted	0.938272
CatBoost_skf_MinMaxScaler_unsmote_weighted	0.936396
CatBoost_skf_MinMaxScaler_smote_unweighted	0.932624
LGBM_skf_MinMaxScaler_unsmote_weighted	0.926916



CatBoost, Smote, Weighted
Highest AUC when Precision > 0.975

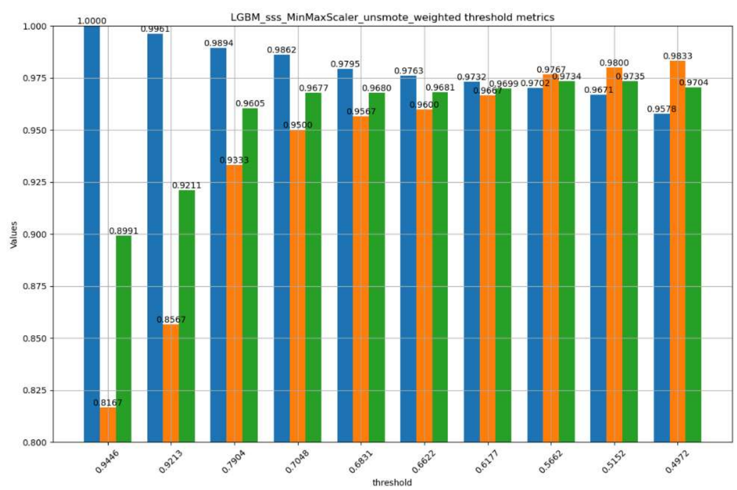
Balanced Performance

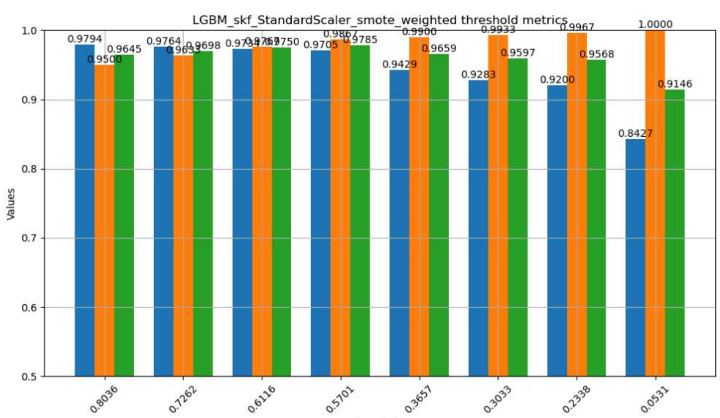
model	recall_at_precision_1	precision_at_recall_1	max_f1_score	optimal_threshold
LGBM_skf_StandardScaler_smote_weighted	0.603333	0.842697	0.978512	0.573696
LGBM_skf_StandardScaler_smote_unweighted	0.603333	0.842697	0.978512	0.573696
CatBoost_sss_MinMaxScaler_unsmote_weighted	0.706667	0.777202	0.976589	0.556067
CatBoost_sss_MaxAbsScaler_unsmote_weighted	0.706667	0.777202	0.976589	0.556067
CatBoost_sss_StandardScaler_unsmote_weighted	0.706667	0.777202	0.976589	0.556067
LGBM_skf_MaxAbsScaler_smote_unweighted	0.570000	0.808625	0.975042	0.655214
LGBM_skf_MaxAbsScaler_smote_weighted	0.570000	0.808625	0.975042	0.655214
LGBM_skf_MinMaxScaler_smote_unweighted	0.570000	0.808625	0.975042	0.655214
LGBM_skf_MinMaxScaler_smote_weighted	0.570000	0.808625	0.975042	0.655214
CatBoost_skf_StandardScaler_smote_weighted	0.706667	0.789474	0.974790	0.718492
CatBoost_skf_MinMaxScaler_smote_weighted	0.706667	0.789474	0.974790	0.718492
CatBoost_skf_MaxAbsScaler_smote_weighted	0.706667	0.789474	0.974790	0.718492
LGBM_sss_MinMaxScaler_unsmote_weighted	0.816667	0.699301	0.973510	0.530220
LGBM_sss_MinMaxScaler_unsmote_unweighted	0.816667	0.699301	0.973510	0.530220
LGBM_sss_MaxAbsScaler_unsmote_weighted	0.816667	0.699301	0.973510	0.530220
LGBM_sss_MaxAbsScaler_unsmote_unweighted	0.816667	0.699301	0.973510	0.530220
CatBoost_tt_MinMaxScaler_unsmote_weighted	0.488136	0.870206	0.973064	0.540427
CatBoost_tt_MaxAbsScaler_unsmote_weighted	0.488136	0.870206	0.973064	0.540427
CatBoost_tt_StandardScaler_unsmote_weighted	0.488136	0.870206	0.973064	0.540427
CatBoost_skf_MinMaxScaler_unsmote_weighted	0.836667	0.773196	0.971993	0.429214
CatBoost_skf_StandardScaler_unsmote_weighted	0.836667	0.773196	0.971993	0.429214
CatBoost_skf_MaxAbsScaler_unsmote_weighted	0.836667	0.773196	0.971993	0.429214
CatBoost_skf_MinMaxScaler_unsmote_unweighted	0.826667	0.750000	0.971901	0.400109
CatBoost_skf_StandardScaler_unsmote_unweighted	0.826667	0.750000	0.971901	0.400109
CatBoost_skf_MaxAbsScaler_unsmote_unweighted	0.826667	0.750000	0.971901	0.400109

Higher Precision: *LGBM, SSS MinMax/MaxAbs*
Either Smote/ Either weights

Despite having lower f1 score by 0.5%, the higher recall with 100% precision: 19.33% of spam allowed at a 95% threshold is more fit for

Setting threshold lower to 52%:
f1: 0.9735
3.39% of real emails classified as spam
2% of spam allowed





Highest f1 score

LGBM, SKF, SMOTE, StandardScaler
57% Threshold: f1: 0.9785
2.95% of real emails in spam box,
1.33% of spam emails in inbox

Feature Importances

CatBoost

Feature	Importance
daren	5.438446
hp	4.468740
attached	4.239964
subject	3.522506
http	3.079105
forwarded	2.452588
nom	2.381251
hanks	2.281593
the	2.183810
will	1.878456
ali	1.814586
gas	1.806800
aren	1.631602
ii	1.361174
mo	1.345133
texas	1.282203
questions	1.281647
deal	1.216775
z	1.101859
th	1.069406
thanks	1.054868
your	0.988475
thank	0.957558
please	0.952757
for	0.894902

LightGBM

Feature	Importance
the	55
will	52
daren	38
z	35
day	32
deal	32
hp	30
questions	29
th	29
forwarded	28
v	27
employee	26
texas	26
mo	26
you	26
s	25
attached	25
p	24
ali	24
money	24
sitara	23
volume	23
r	23
ii	23
is	22