PHISHING WEBSITES DETECTION

AI for Cybersecurity (931II)
M.Sc. Cybersecurity

Paolo Bernardi (660944)

2024-02-14

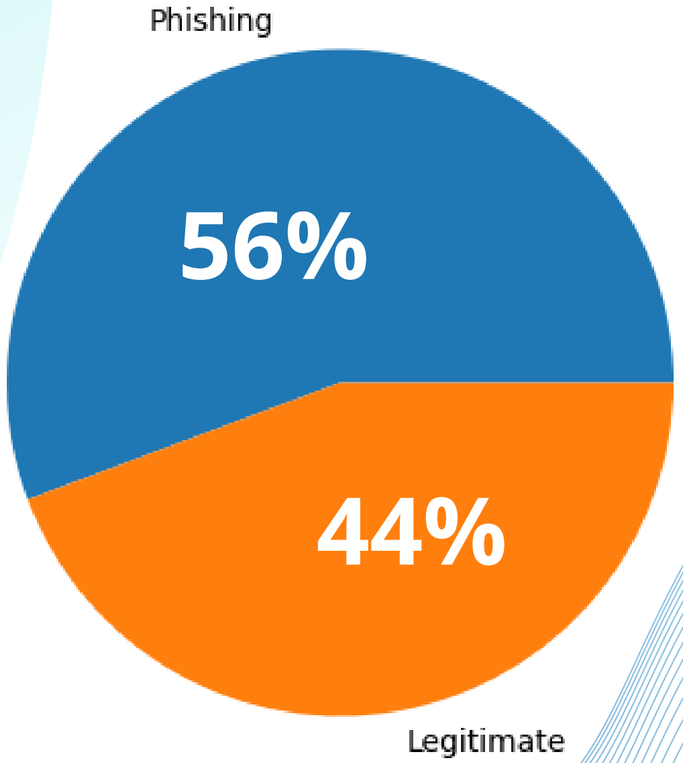
GOAL

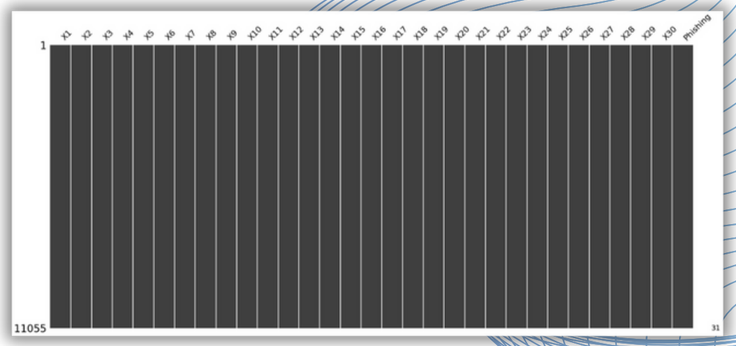
- Find the most efficient among **5 different classifiers** to detect phishing websites
- Comparing dataset **with** and **without duplicates**, if it makes sense
- Comparing 2 different feature reduction strategies: **Variance Threshold** and **L1-based Linear SVC**
- Comparing classifiers with **default and with optimized hyperparameters**
- Acceptable accuracy: around **95%**

1. **K-Nearest Neighbors**
2. **Linear SVC**
3. **Logistic Regression**
4. **Decision Tree**
5. **Random Forest**

DATASET

- Phishing Websites Features
- Mohammad, R. & McCluskey, L. (2015, University of Huddersfield)
- UCI Machine Learning Repository https://doi.org/10.24432/C52W2X
- ARFF file
- **30** features
- **11055** rows (**6157** Phishing, **4898** Legitimate)
- Nominal values (Y/N, Y/N/Unsure)
- Already **normalized** data
- No null values





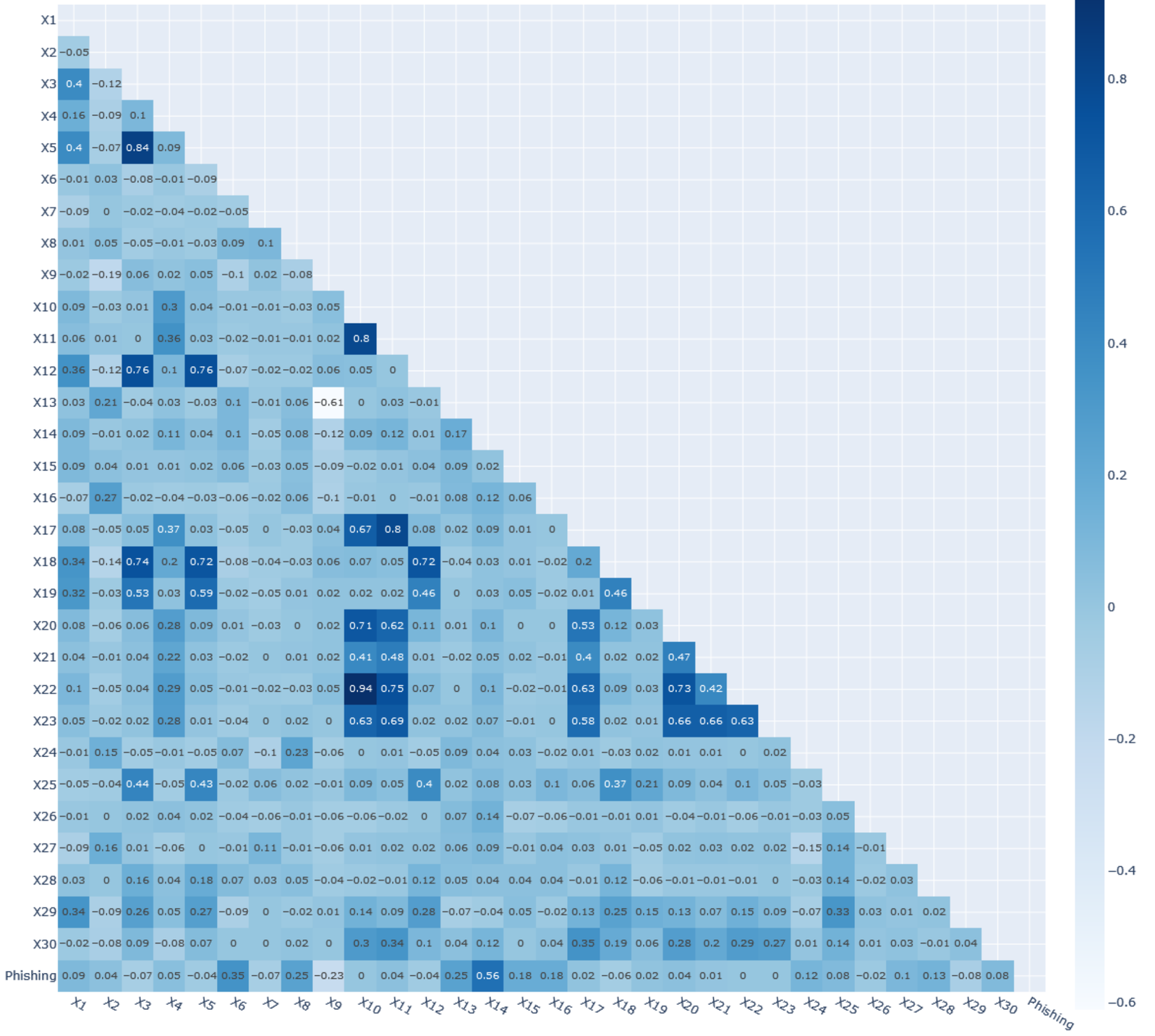
DATASET

Renamed for compact display

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	...	X22	X23	X24	X25	X26	X27	X28	X29	X30	Phishing
0	-1	1	1	1	-1	-1	-1	-1	-1	1	...	1	1	-1	-1	-1	-1	1	1	-1	-1
1	1	1	1	1	1	-1	0	1	-1	1	...	1	1	-1	-1	0	-1	1	1	1	-1
2	1	0	1	1	1	-1	-1	-1	-1	1	...	1	1	1	-1	1	-1	1	0	-1	-1
3	1	0	1	1	1	-1	-1	-1	1	1	...	1	1	-1	-1	1	-1	1	-1	1	-1
4	1	0	-1	1	1	-1	1	1	-1	1	...	-1	1	-1	-1	0	-1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11050	1	-1	1	-1	1	1	1	1	-1	-1	...	-1	-1	1	1	-1	-1	1	1	1	1
11051	-1	1	1	-1	-1	-1	1	-1	-1	-1	...	-1	1	1	1	1	1	1	-1	1	-1
11052	1	-1	1	1	1	-1	1	-1	-1	1	...	1	1	1	1	1	-1	1	0	1	-1
11053	-1	-1	1	1	1	-1	-1	-1	1	-1	...	-1	1	1	1	1	-1	1	1	1	-1
11054	-1	-1	1	1	1	-1	-1	-1	1	1	...	1	1	-1	1	-1	-1	-1	1	-1	-1

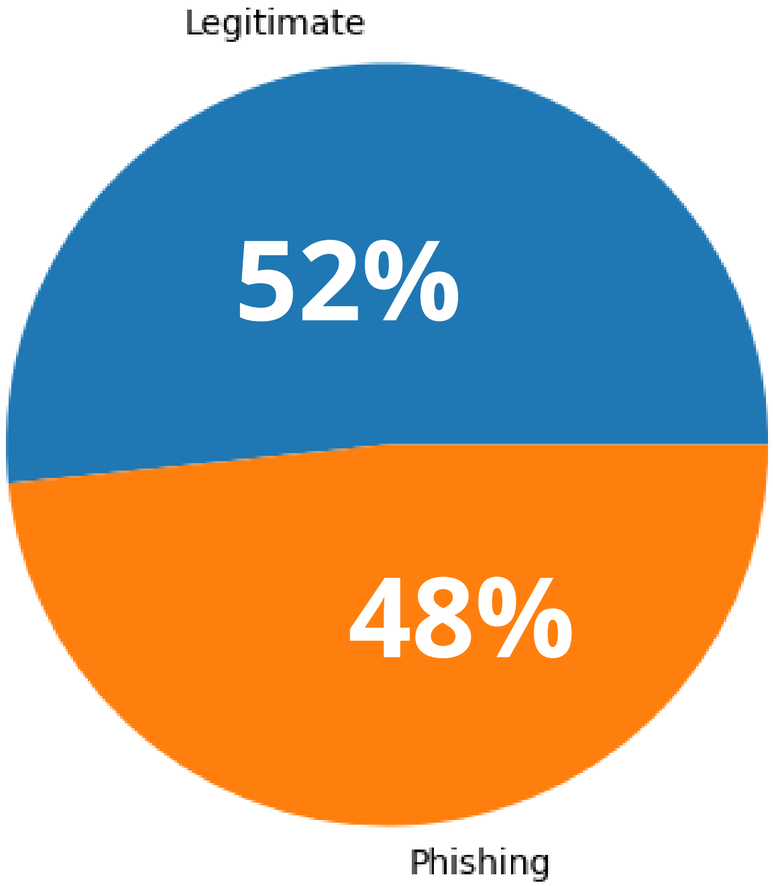
-1 = No
0 = Maybe
1 = Yes

CORRELATION MATRIX



DUPLICATES REMOVAL

- The dataset becomes even more balanced
- **5849** rows (**3019** Phishing, **2830** Legitimate)
- Removed **47%** of the rows (**54%** Phishing, **38%** Legitimate)

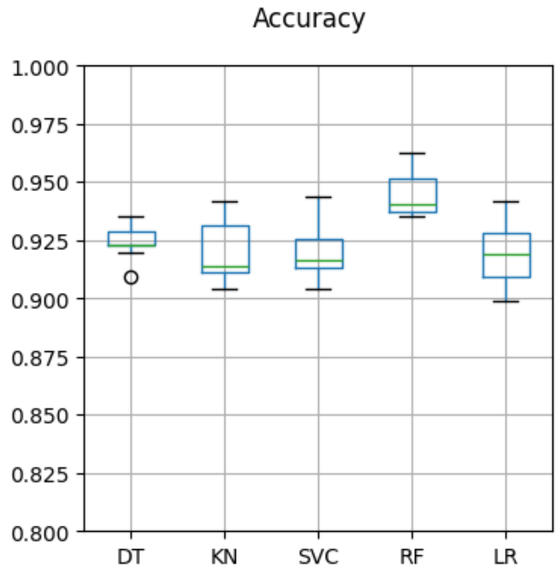


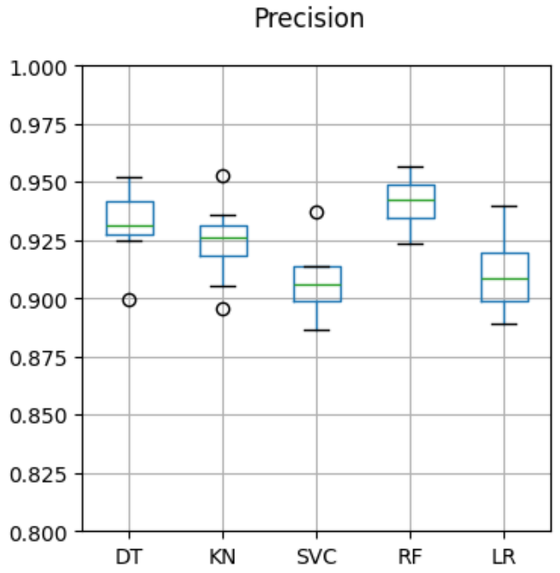
Does it make sense?

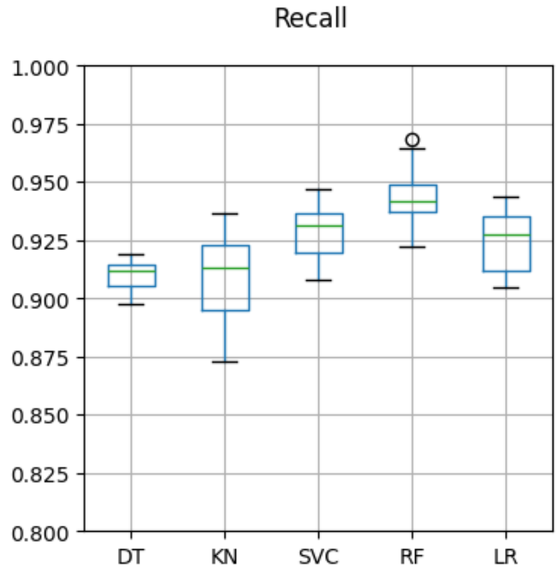
- **Here duplicates don’t seem to have a special meaning**
- **They will probably lead to an overfitted and biased result**

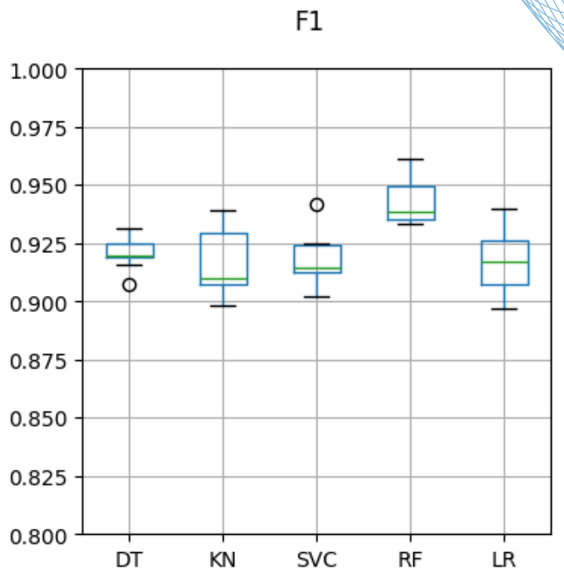
NO DUPLICATES & VARIANCE THRESHOLD → 4 pruned features

BEFORE HYPERPARAMETER OPTIMIZATION









RANDOM FOREST OVERALL WIN

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.922893	0.919813	0.919130	0.942212	0.919131
test_precision	0.932207	0.924363	0.906646	0.937912	0.909727
test_recall	0.906714	0.908834	0.928622	0.943110	0.924735
test_f1	0.919235	0.916406	0.917455	0.940461	0.917130

$\alpha = 0.5$

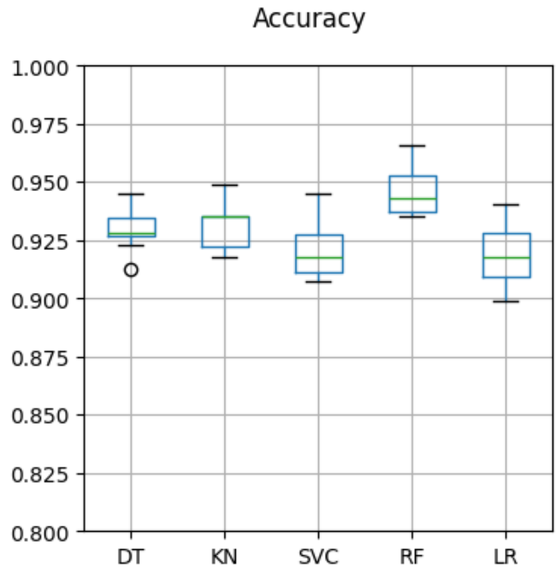
Null Hypothesis

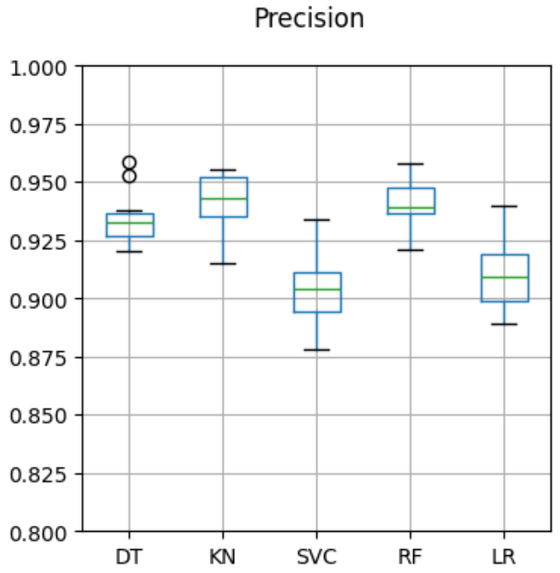
Wilcoxon test

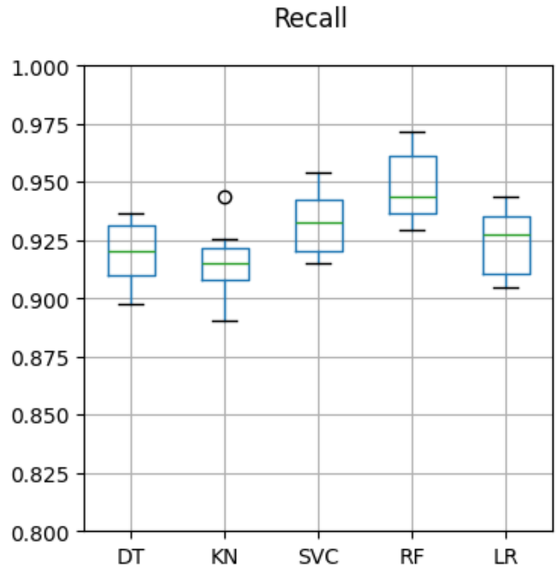
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.944045	0.173071	0.090488	0.858955
SVC	DT	0.207021	0.009766	0.007632	0.322266
DT	KN	0.375000	0.232422	0.674047	0.492188
RF	DT	**0.001953**	0.105469	**0.001953**	**0.001953**
RF	KN	**0.001953**	0.064453	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	0.003906	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

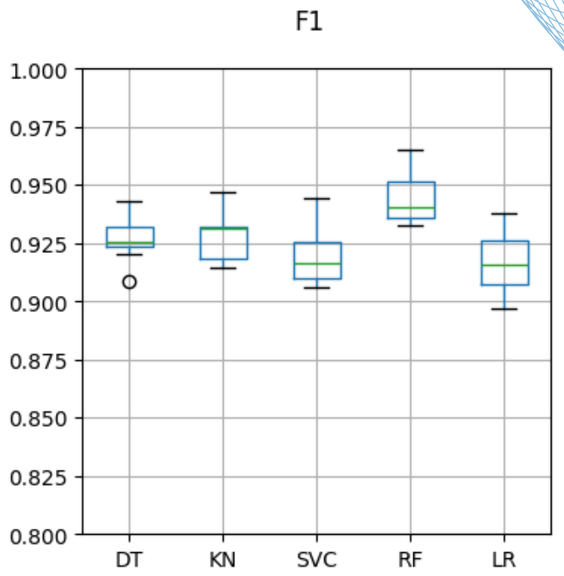
NO DUPLICATES & VARIANCE THRESHOLD → 4 pruned features

AFTER HYPERPARAMETER OPTIMIZATION









RF IMPROVES SLIGHTLY AND STILL WINS, BUT NOT COMPLETELY

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.929733	0.931269	0.920155	0.946146	0.918789
test_precision	0.934744	0.941591	0.904971	0.941425	0.909668
test_recall	0.919081	0.914841	0.933216	0.947703	0.924028
test_f1	0.926766	0.927944	0.918791	0.944504	0.916751

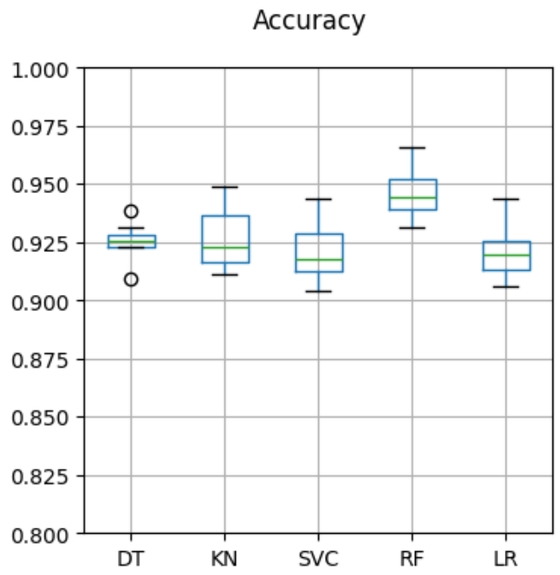
$\alpha = 0.5$ Null Hypothesis

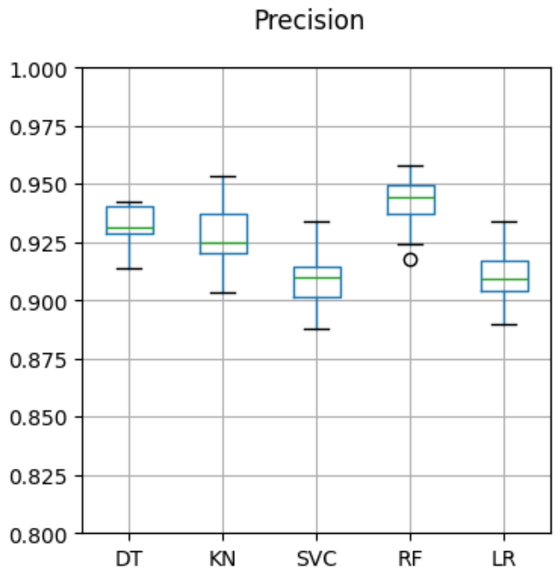
Wilcoxon test

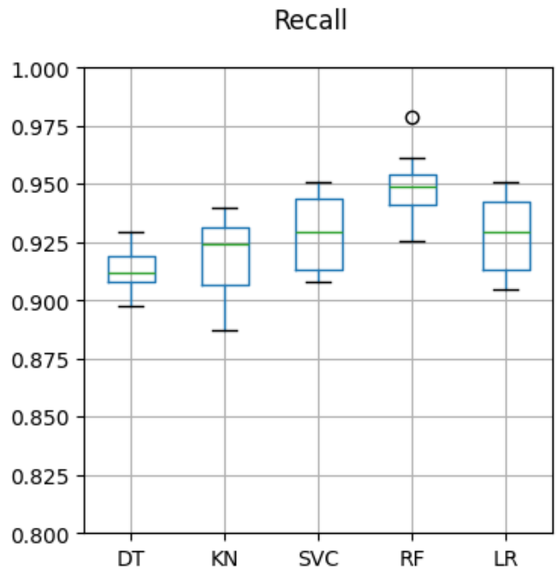
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.625000	0.160156	0.043640	0.375000
SVC	DT	0.027344	0.003906	0.019531	0.160156
DT	KN	0.845703	0.232422	0.556641	0.845703
RF	DT	0.007686	0.130859	0.003906	0.003906
RF	KN	**0.001953**	1.000000	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	0.023151	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

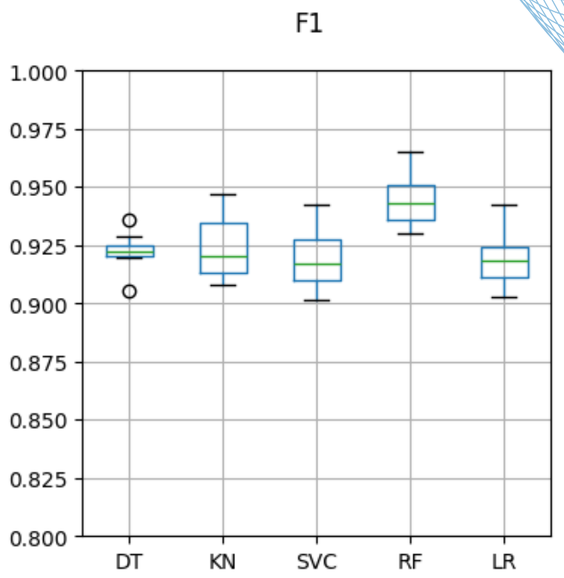
NO DUPLICATES & L1-BASED LINEAR SVC → 3 pruned features

BEFORE HYPERPARAMETER OPTIMIZATION









L1 FEATURE SELECTION ARGUABLY PERFORMS SLIGHTLY BETTER...

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.925628	0.925969	0.921524	0.946316	0.921524
test_precision	0.931613	0.927378	0.910472	0.941194	0.911875
test_recall	0.913428	0.919081	0.929329	0.948410	0.927562
test_f1	0.922392	0.923137	0.919721	0.944720	0.919578

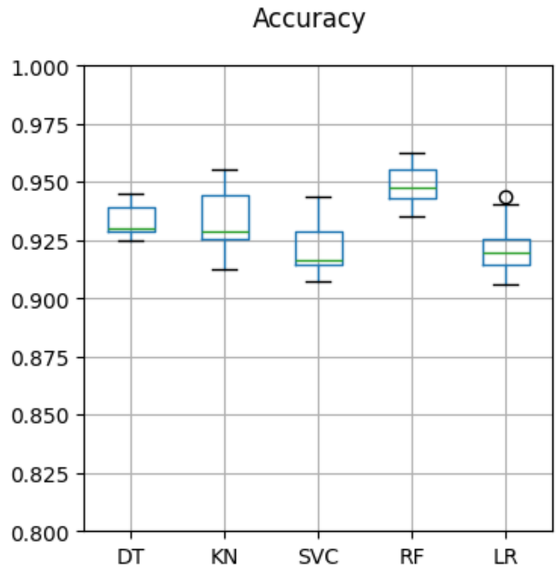
$\alpha = 0.5$ Null Hypothesis

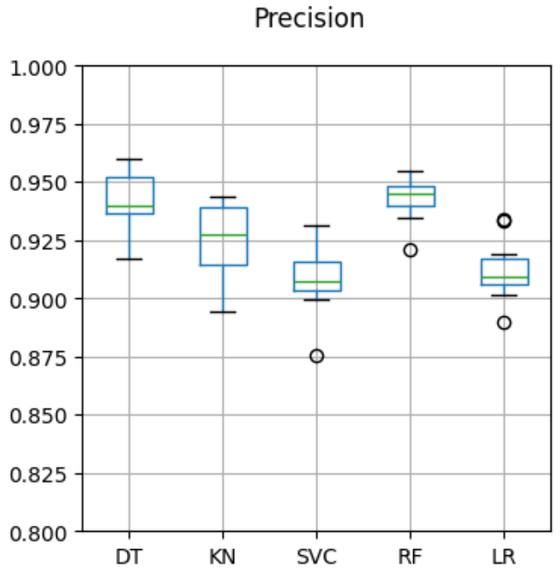
Wilcoxon test

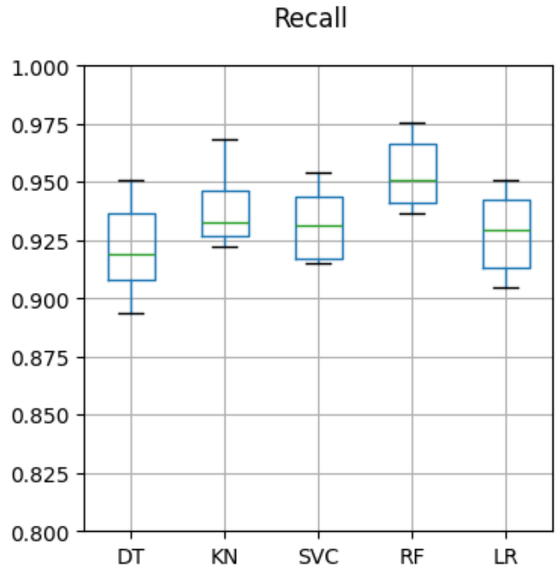
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.904776	0.678402	0.058782	0.767097
SVC	DT	0.275391	**0.001953**	0.024265	0.695312
DT	KN	1.000000	0.193359	0.313088	1.000000
RF	DT	**0.001953**	0.013672	**0.001953**	**0.001953**
RF	KN	**0.001953**	0.003906	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	0.003906	**0.001953**	**0.001953**

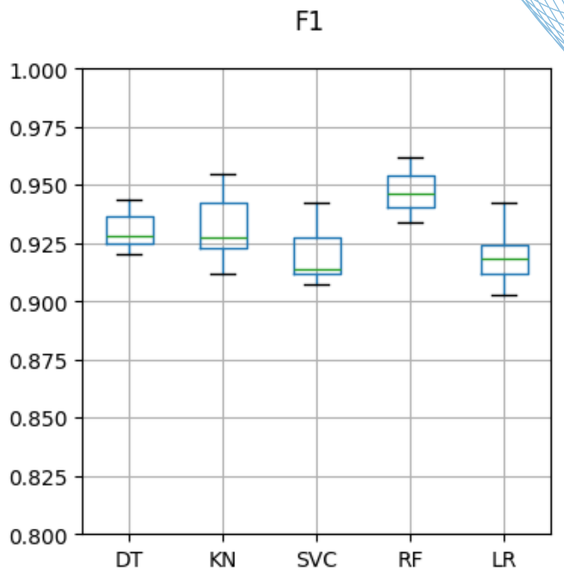
NO DUPLICATES & L1-BASED LINEAR SVC → 3 pruned features

AFTER HYPERPARAMETER OPTIMIZATION









... AND HYPERPARAMETER OPTIMIZATION IMPROVES IT A LITTLE BIT

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.933666	0.933150	0.921181	0.949394	0.921695
test_precision	0.941419	0.924878	0.907919	0.943080	0.912190
test_recall	0.920495	0.938163	0.931802	0.953004	0.927562
test_f1	0.930643	0.931423	0.919620	0.947954	0.919737

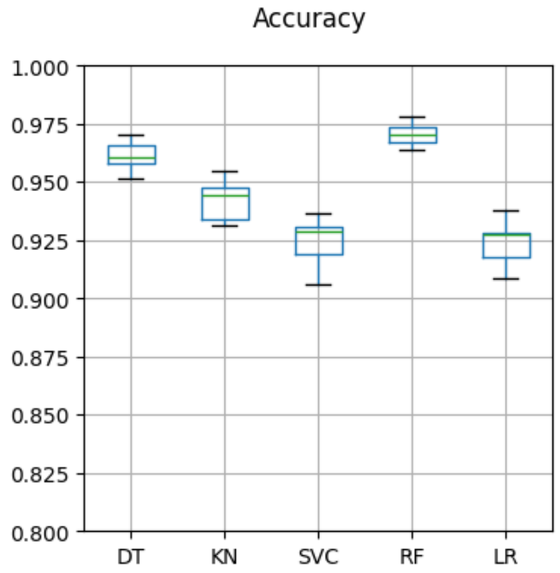
$\alpha = 0.5$ Null Hypothesis

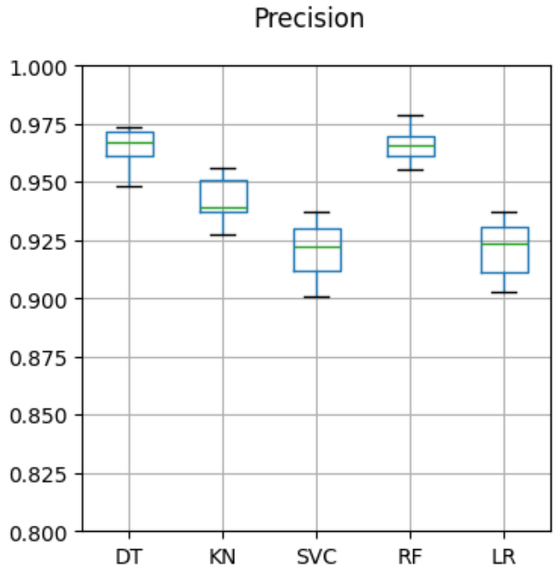
Wilcoxon test

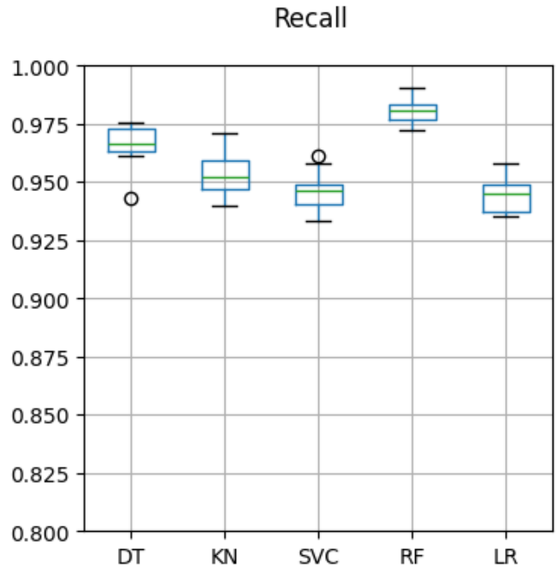
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.634992	0.160156	0.091690	0.921875
SVC	DT	0.009766	**0.001953**	0.013672	0.019531
DT	KN	0.812380	0.013672	0.027344	1.000000
RF	DT	**0.001953**	0.625000	**0.001953**	**0.001953**
RF	KN	**0.001953**	**0.001953**	0.011311	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

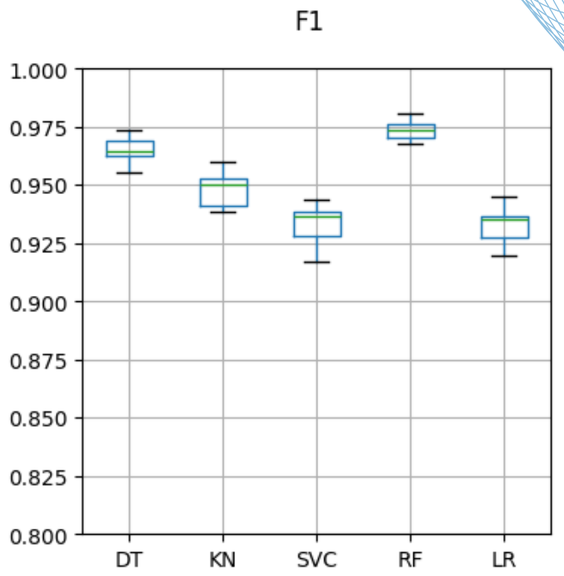
DUPLICATES & VARIANCE THRESHOLD → 4 pruned features

BEFORE HYPERPARAMETER OPTIMIZATION









PERFORMANCE WITH DUPLICATES IS SIGNIFICANTLY BETTER...

$\alpha = 0.5$

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.961376	0.941746	0.924739	0.970240	0.924377
test_precision	0.965038	0.942609	0.921063	0.966599	0.921549
test_recall	0.965730	0.953545	0.946076	0.980509	0.944778
test_f1	0.965334	0.948003	0.933362	0.973481	0.932978

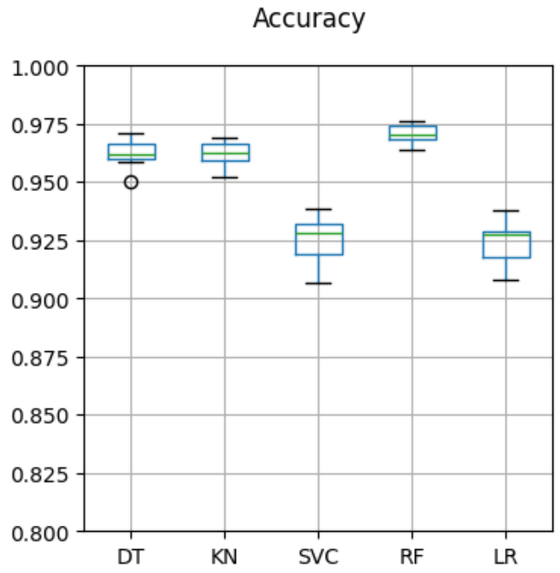
Null Hypothesis

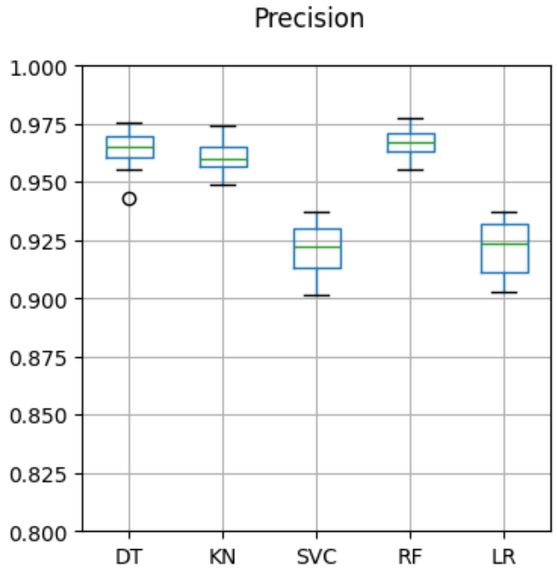
Wilcoxon test

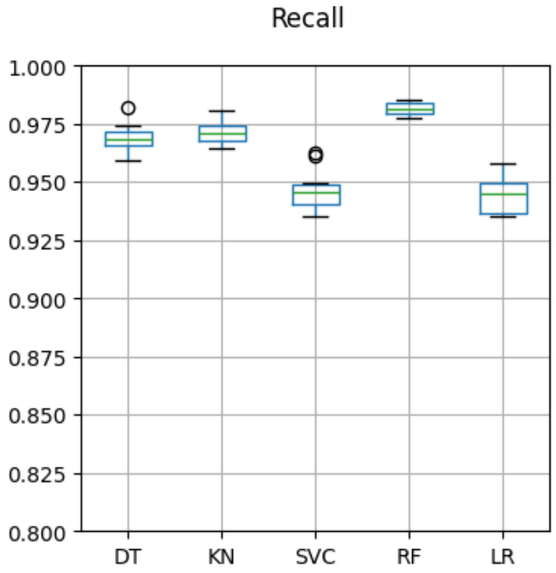
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.695312	0.695312	0.232508	0.492188
SVC	DT	**0.001953**	**0.001953**	0.003906	**0.001953**
DT	KN	**0.001953**	**0.001953**	0.013672	**0.001953**
RF	DT	**0.001953**	0.375000	**0.001953**	**0.001953**
RF	KN	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

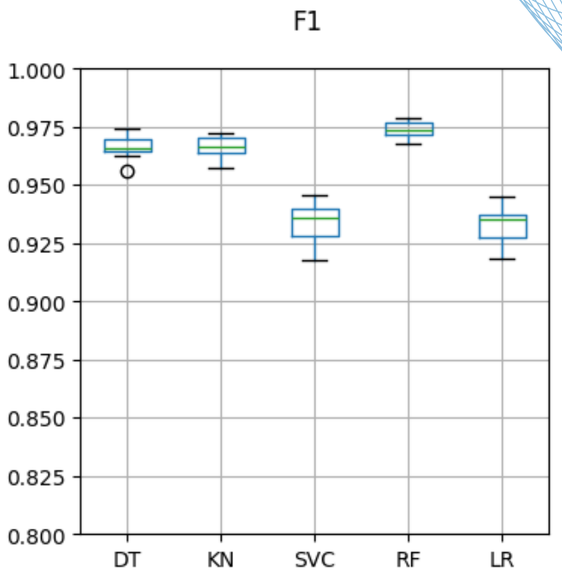
DUPLICATES & VARIANCE THRESHOLD → 4 pruned features

AFTER HYPERPARAMETER OPTIMIZATION









... BUT WHAT ABOUT BIAS AND OVERFITTING?

$\alpha = 0.5$

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.962371	0.962280	0.925191	0.970964	0.924287
test_precision	0.963752	0.961473	0.921385	0.967073	0.921674
test_recall	0.968979	0.971251	0.946563	0.981321	0.944453
test_f1	0.966323	0.966316	0.933764	0.974132	0.932882

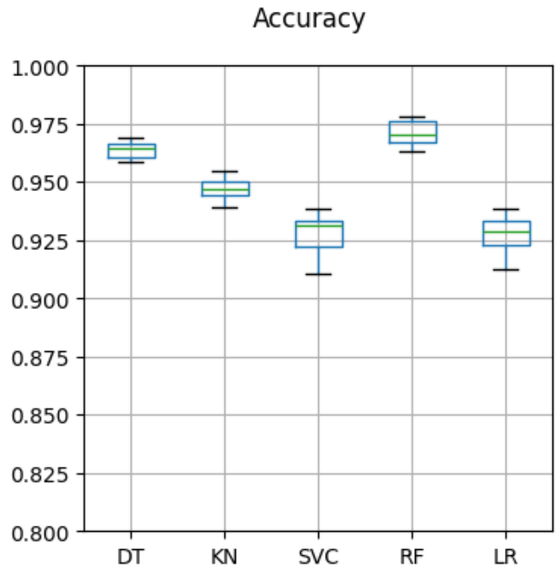
Null Hypothesis

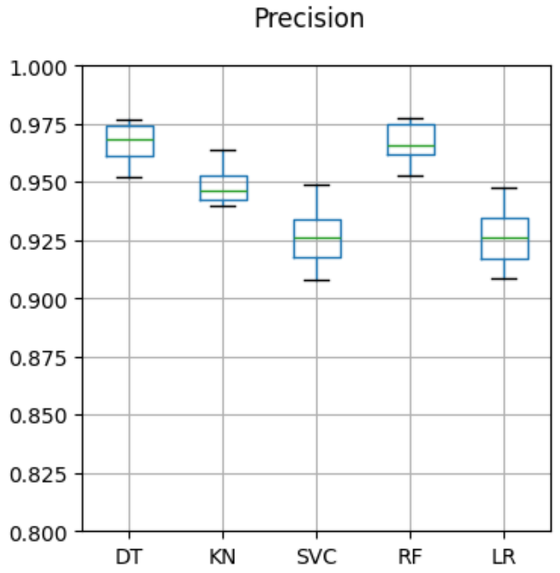
Wilcoxon test

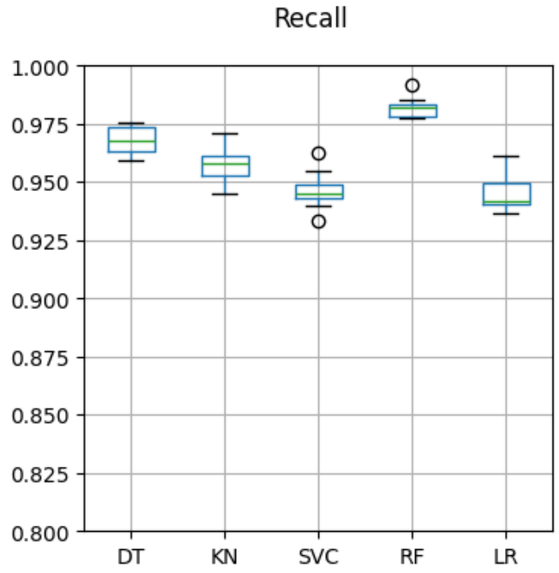
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.375000	0.845703	0.108185	0.232422
SVC	DT	**0.001953**	**0.001953**	**0.001953**	**0.001953**
DT	KN	1.000000	0.160156	0.192127	0.769531
RF	DT	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	KN	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

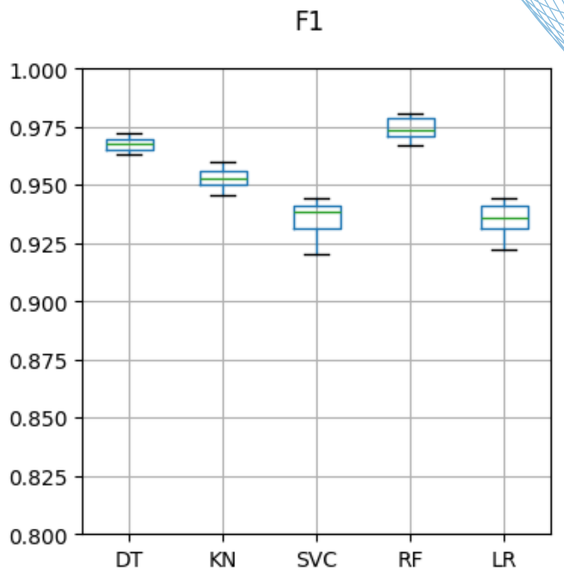
DUPLICATES & L1-BASED LINEAR SVC → 3 pruned features

BEFORE HYPERPARAMETER OPTIMIZATION









AS ALREADY WITNESSED, L1 FEATURE SELECTION IS BETTER

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.963637	0.947264	0.927906	0.971055	0.927453
test_precision	0.966852	0.948748	0.926089	0.966660	0.926437
test_recall	0.968004	0.957121	0.946237	0.981971	0.944938
test_f1	0.967384	0.952871	0.936001	0.974232	0.935541

$\alpha = 0.5$

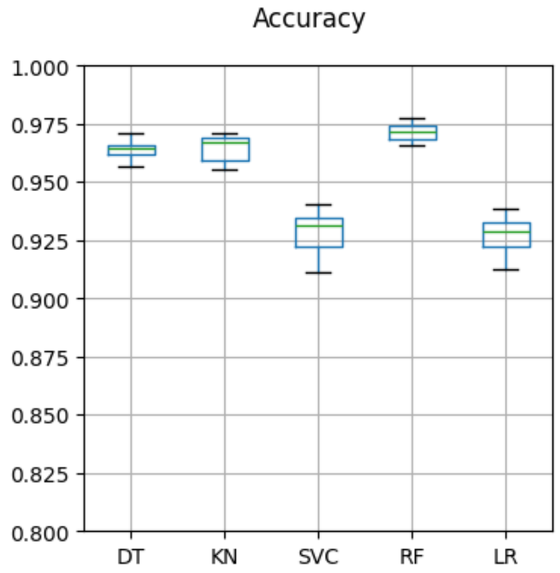
Null Hypothesis

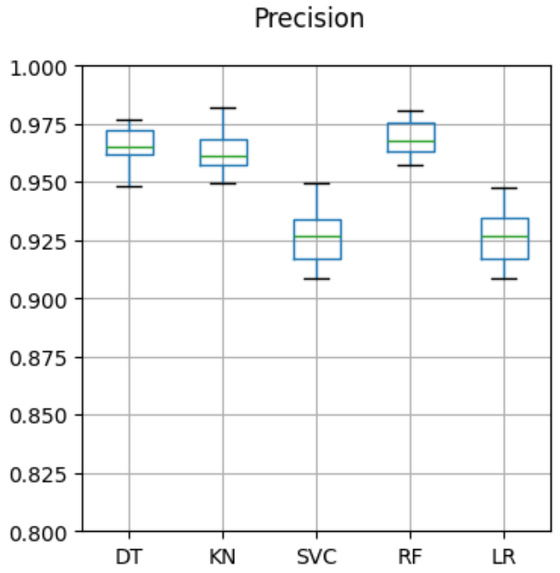
Wilcoxon test

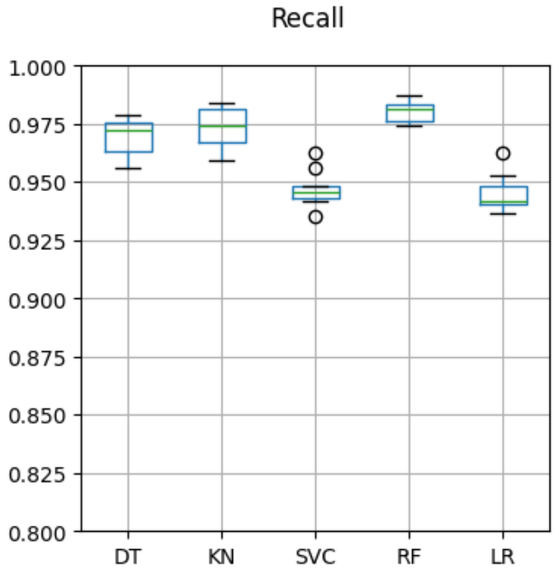
Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.751830	0.625000	0.431641	0.322266
SVC	DT	**0.001953**	**0.001953**	**0.001953**	**0.001953**
DT	KN	**0.001953**	**0.001953**	0.005859	**0.001953**
RF	DT	**0.001953**	0.695312	**0.001953**	**0.001953**
RF	KN	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

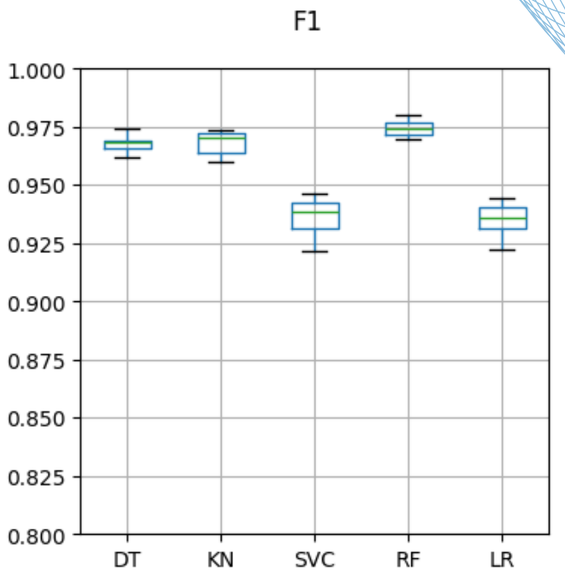
DUPLICATES & L1-BASED LINEAR SVC → 3 pruned features

AFTER HYPERPARAMETER OPTIMIZATION









AND HYPERPARAMETER OPTIMIZATION IMPROVES THE RESULTS A LITTLE BIT

Performance

Cross Validation with Stratified K-Fold, 10 splits

	DT	KN	SVC	RF	LR
test_accuracy	0.964089	0.964632	0.928268	0.971416	0.927363
test_precision	0.965684	0.963313	0.926135	0.968781	0.926292
test_recall	0.970115	0.973687	0.946887	0.980347	0.944938
test_f1	0.967839	0.968424	0.936347	0.974502	0.935466

$\alpha = 0.5$

Null Hypothesis

Wilcoxon test

Cls 1	Cls 2	Accuracy p-value	Precision p-value	Recall p-value	F1 p-value
SVC	LR	0.231073	0.695312	0.154407	0.275391
SVC	DT	**0.001953**	**0.001953**	**0.001953**	**0.001953**
DT	KN	0.625000	0.130859	0.205903	0.492188
RF	DT	**0.001953**	0.048828	**0.001953**	**0.001953**
RF	KN	**0.001953**	0.003906	**0.001953**	**0.001953**
RF	SVC	**0.001953**	**0.001953**	**0.001953**	**0.001953**
RF	LR	**0.001953**	**0.001953**	**0.001953**	**0.001953**

RESULTS SUMMARY

EVALUATED ALGORITHMS

1. Decision Tree
2. K-Nearest Neighbors
3. Linear SVC
4. Random Forest
5. Logistic Regression

DEGREES OF FREEDOM

2 feature selection strategies: Variance Threshold, L1-based Linear SVC

With and without **hyperparameter optimization**

With and without **duplicates**

WINNING MODEL

Random Forest

It wins hands down in every configuration. If processor usage is a constraint, DT is a suitable alternative.

FEATURE SELECTION

L1-based Linear SVC

This feature selection strategy provides the best results, but most likely just because it removes less features.

DUPLICATES RETENTION

Remove Duplicates

Since duplicates have no special meaning in this specific case, the performance gains obtained by keeping them are at high risk of bias and overfitting.

REAL-TIME ANALYSIS

Metaverse-ready UI

```
D:\github\website_phishing_checker>python check.py

--- PHISHING WEBSITE CHECKER ---

URL = https://webmail-109686.weeblysite.com/

Analyzing website...
[  0%] URL analysis...
[ 10%] SSL certificate...
[ 20%] Expiry date...
[ 30%] WHOIS...
[ 45%] Domain registration age...
[ 60%] Website traffic...
[ 75%] Website ranking...
[ 90%] Statistical report...
[100%] Done.

  Having IP Address = NO
  URL Length = NO
  Shortening Service = NO
  Having At Symbol = NO
  Double Slash Redirecting = NO
  Prefix Suffix = YES
  Having Sub Domain = YES
  SSL Final State = NO
  Domain Registration Length = YES
  Port = NO
  HTTPS Token = NO
  Request URL = YES
  URL of Anchor = YES
  Links in Tags = YES
  SFH = NO
  Submitting to Email = NO
  Abnormal URL = YES
  Redirect = NO
  On Mouseover = NO
  Right Click = NO
  Age of Domain = NO
  DNS Record = NO
  Web Traffic = NO
  Page Rank = NO
  Google Index = NO
  Links Pointing to Page = YES
  Statistical Report = YES

Loading classifier...
ACHTUNG: POSSIBLE PHISHING ATTEMPT
```

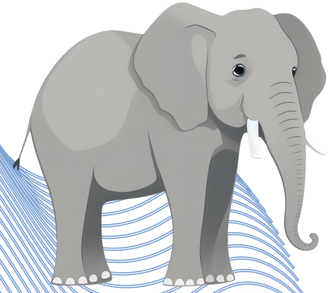
Command line Python tool that:

1. Extracts the features of interest from a user-specified website
2. Loads the classifier
3. Classifies the website as phishing / NOT phishing

PHISHING

WEBSITE CHECKER

ON THE DATASET, AGAIN



Readily actionable for data analysis & experimentation

The dataset is clean, balanced and normalized

BUT

The documentation is lacking w.r.t. features extraction procedures

Extending the dataset and reusing it in real-time apps is challenging

EXAMPLES

1.3.3. Disabling Right Click

Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. This feature is treated exactly as "Using onMouseOver to hide the Link". Nonetheless, for this feature, we will search for event "event.button==2" in the webpage source code and check if the right click is disabled.

1+1==event.button?

1.4.3. Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information

Dismissed in 2022

NEXT STEPS

What could the future of this work be?

1

Create the dataset anew

Reuse the features suggested by the authors, perhaps with a few adjustments. More importantly, **provide the code for feature extraction** and, possibly, the list of websites used for the dataset.

2

Re-run the data analysis

Verify whether the conclusions of this work hold or, perhaps, with a new dataset there are significant shifts.

3

Create a robust real-time app

It should be fairly easy, by combining the code for feature extraction and the data analysis results.

THANK YOU

FOR YOUR ATTENTION