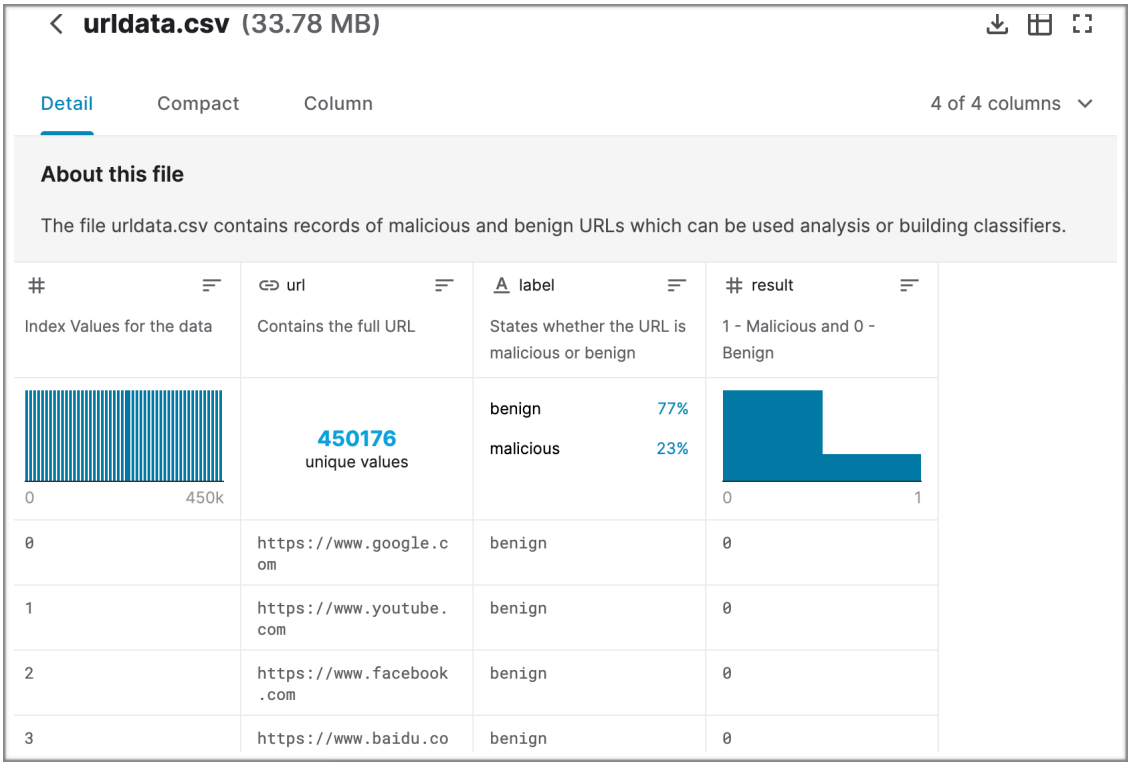
Malicious URL Detection

By lexical features - DS4CS Final Project

National Chengchi University 108 semester by Jason Hung

I. Dataset

https://www.kaggle.com/siddharthkumar25/malicious-and-benign-urls



urldata.csv (33.78 MB)

Detail | Compact | Column | 4 of 4 columns

About this file

The file urldata.csv contains records of malicious and benign URLs which can be used analysis or building classifiers.

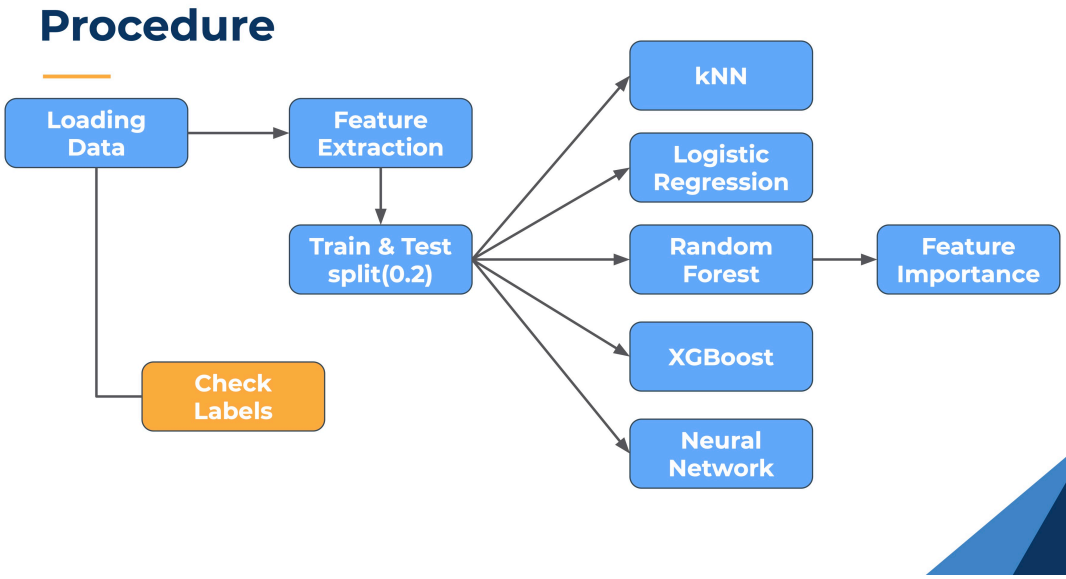
#	url	label	result
Index Values for the data	Contains the full URL	States whether the URL is malicious or benign	1 - Malicious and 0 - Benign
0 – 450k	450176 unique values	benign 77%, malicious 23%	0 – 1
0	https://www.google.com	benign	0
1	https://www.youtube.com	benign	0
2	https://www.facebook.com	benign	0
3	https://www.baidu.co	benign	0

Datasets 選擇Kaggle上 Siddharth Kumar 提供的，主要有兩個欄位，URL跟Label，是已經將每個URL標註為Benign與Malicious的資料集。總共筆數為450,176，Benign與Malicious的比例約為3:1，稍微不平衡一點，但還在可以接受範圍內。所以使用這個資料集的第一個想法，是需要做特徵抽取(Feature Extraction)，從URL做出能夠代表這個URL的特徵。在檢查資料集內是否有缺失值(NA)後，發現沒有缺失值，表示這資料集算完整的。

II. Problem

- Binary Classification 二元分類問題
- Try to detect malicious url by lexical features 嘗試透過從URL中擷取文字特徵來找到惡意連結

III. Procedure



我的主要流程為載入資料後，先檢查 Label 跟 Result欄位有沒有標錯的情況，之後進行特徵擷取，因為沒有測試資料，所以從資料拆80%訓練資料集與20%的測試資料集。再來進行分類模型的建模預測，在這裡我挑選五種分類模型，分別是kNN、羅吉斯回歸、隨機森林、XGBoost以及搭建簡單神經網路進行預測，最後從 tree-based的分類模型中看特徵重要度，找出影響程度比較大的特徵進行解釋。

	Result	
Label	0	1
benign	345738	0
malicious	0	104438

將Result、Label 畫交叉表，看有沒有benign標成1或是malicious標成0的情況，從表中得知沒有這類情況，encoding完的label是標注正確的，這邊就不去檢查每個URL標註為好與壞的正確性與否，假設這資料集是經過驗證的標註。

IV. Feature Extraction

如同老師於報告期間所說，如果沒有對URL有很深的了解，會不知道怎麼從URL擷取重要的特徵，所以需要專家的幫助。於是我去找了相關文獻，根據 “Malicious URL filtering — A big data application” 這篇論文中提到了許多從URL抓特徵的方法，我採取了下列幾項特徵作為訓練：

- Top-Level Domain (tld)
- Length of URL
- Length of Domain name
- Special Characters count
- Digits count
- “https” count

再來進行資料探索，觀察這些變數的一些統計數值與分佈，可以看出資料大多呈現右偏狀態，之後可以取log讓分佈較近於常態去實驗結果是否較佳。

Distribution

Name	Distributio	Center	Dispersion	Min.	Max.	Missing
https_count		0.82	0.47	0	4	0 (0%)
domain_toke...		3.14	0.21	1	19	0 (0%)
domain_len		19.26	0.33	0	235	0 (0%)
digits		3.94	2.13	0	631	0 (0%)
special_count		7.04	0.68	0	170	0 (0%)
url_len		59.37	0.59	8	2314	0 (0%)
ntld		1.06	0.23	1	4	0 (0%)

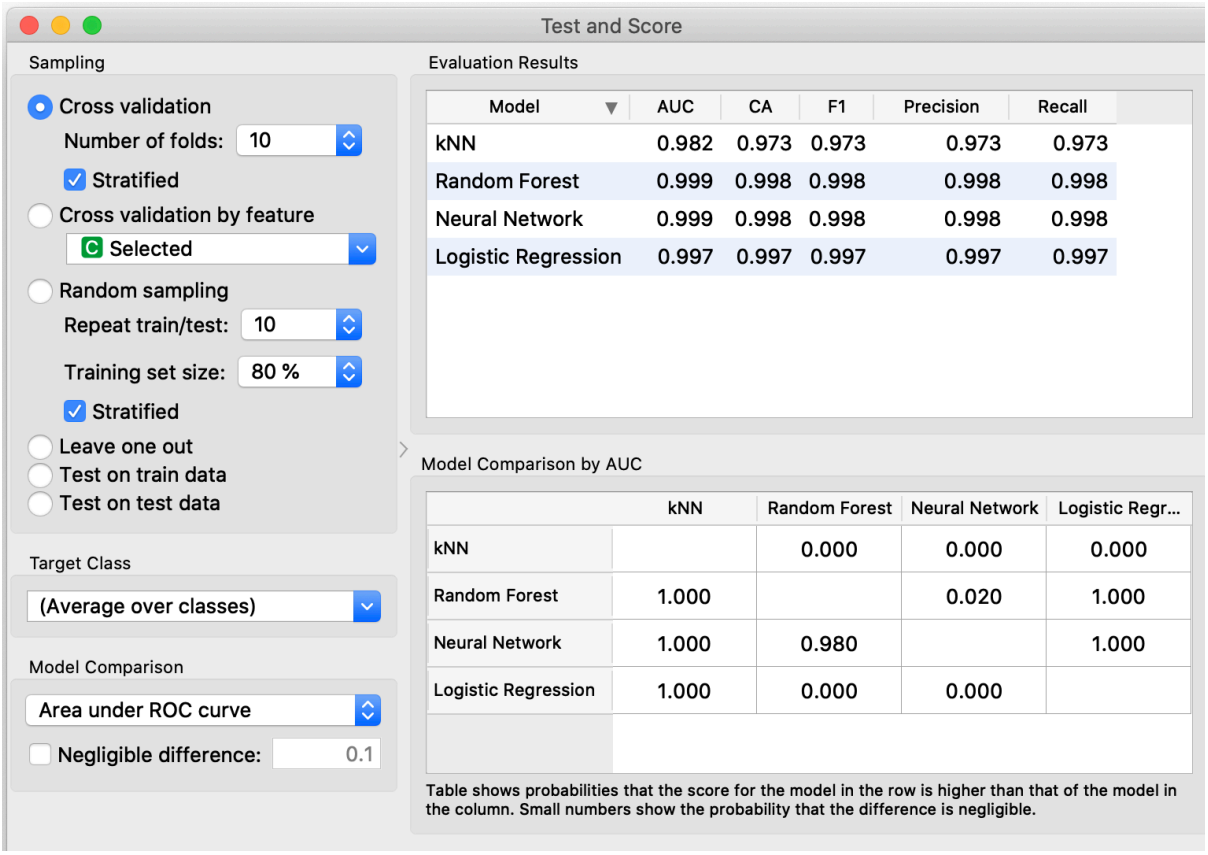
變數之間的關係

V. Modeling

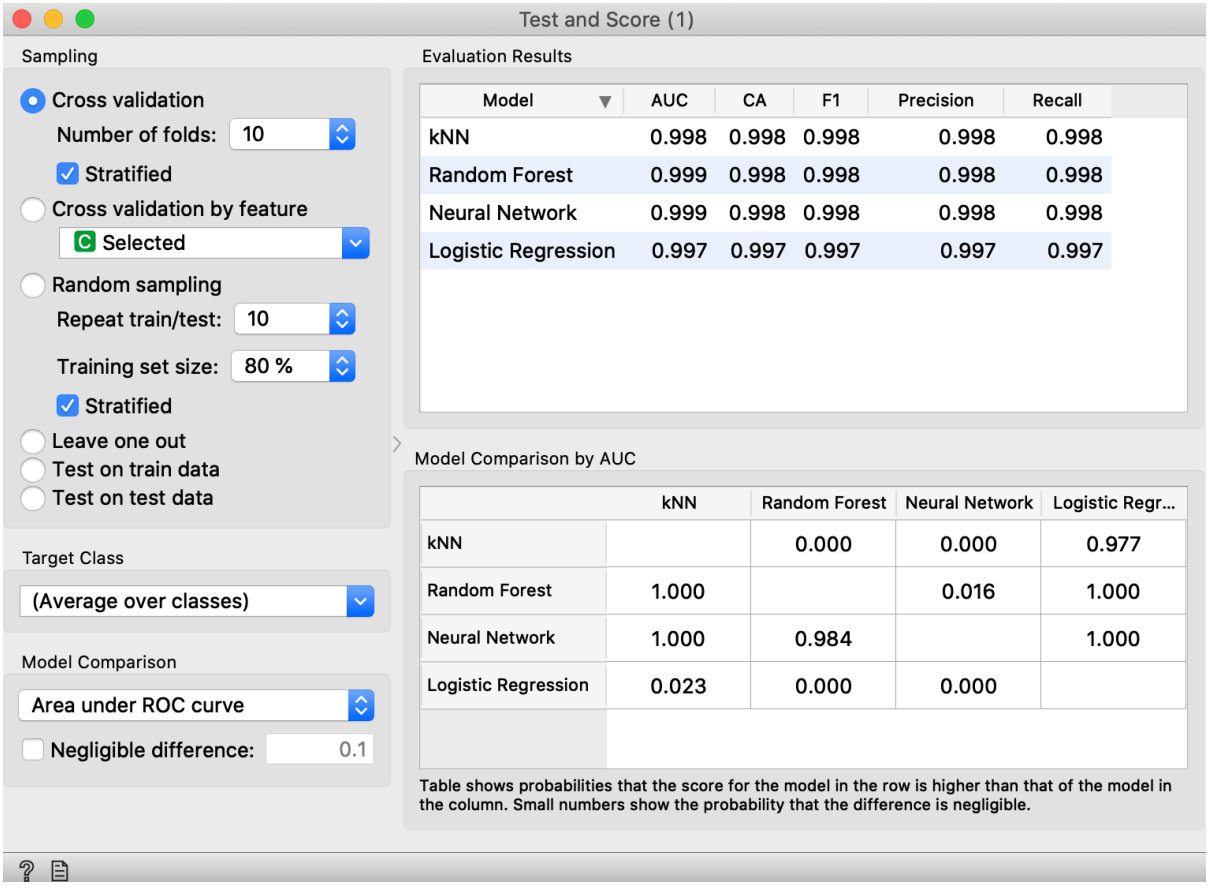
在這邊我嘗試了五種分類模型進行實驗，分別為以下：

- kNN
- Logistic Regression
- Random Forest
- XGBoost
- Neural Network (64 neurons; ReLU)

並除了 tree-based 模型外，進行有無將資料標準化的效果比較，以結果來說標準化後kNN有較明顯的提升，其他模型則相差無幾。

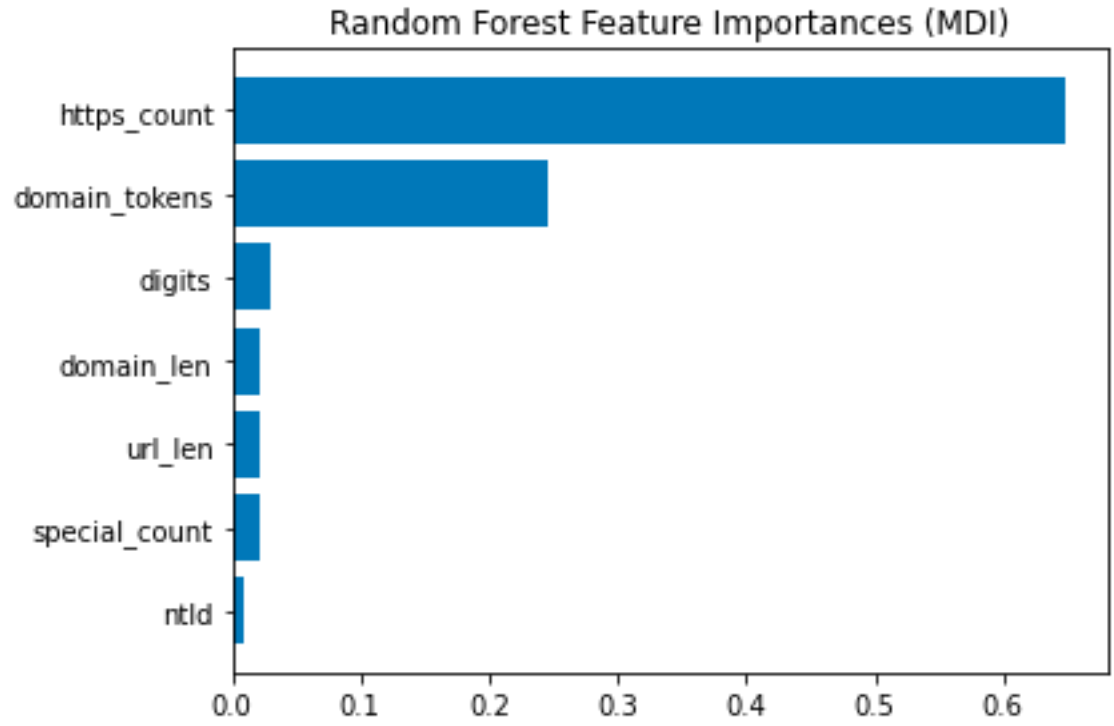


上圖為標準化前，下圖為標準化後的模型比較 (Orange無XGboost模型，故這裡不列出)

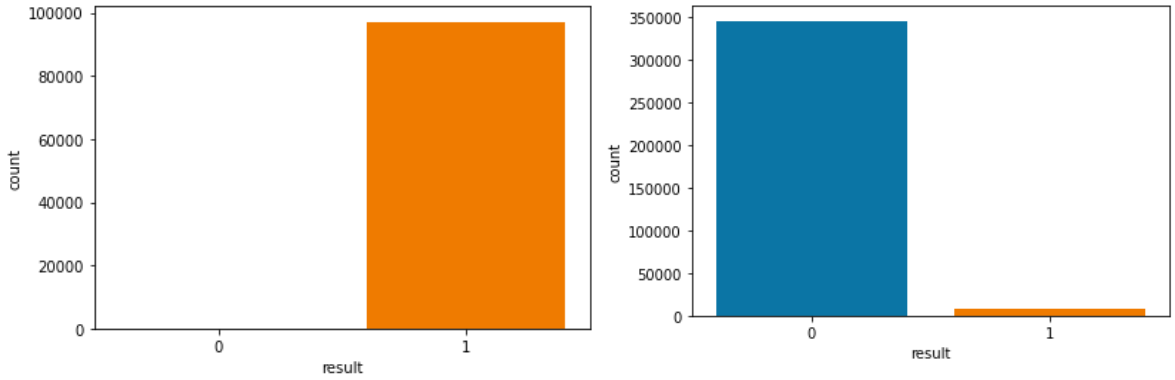


	Logistic Regression	kNN	Random Forest	XGboost	Neural Network
Accuracy	0.9842	0.982	0.9921	0.9841	0.999
F1-score	0.98	0.973	0.99	0.98	0.998
0 Benign	0.99	-	0.99	0.99	-
1 Malicious	0.97	-	0.99	0.97	-

最後我將實驗結果整理成上表，以實驗結果以Random Forest表現為最佳，其次為Neural Network，但兩者的差異非常小。整體來說整體分類的準確度異常的好，代表這些特徵能有效分出是不是惡意連結。接下來我透過隨機森林的特徵重要度畫圖，看哪個特徵是分類的重要依據。



發現計算https是最重要的特徵，接下來就想多觀察這個特徵，於是將htts==0跟https>0的數量畫出來。發現在這個資料集中，只要有https都會是安全的連結，沒有https就是惡意連結，後面也畫出與目標變數的相關係數，也呈現很大的負相關。見下圖（左邊為非https連結，右邊為https連結。



接下來將threshold設為1(https>1)，就全是惡意連結，表示只有一個https的加密連結很多都是安全連結，但是只要大於1個以上就都是惡意連結，下圖為惡意連結的樣子，會

```
1 data[data.https_count>1].result.value_counts()
```

```
1    189
Name: result, dtype: int64
```

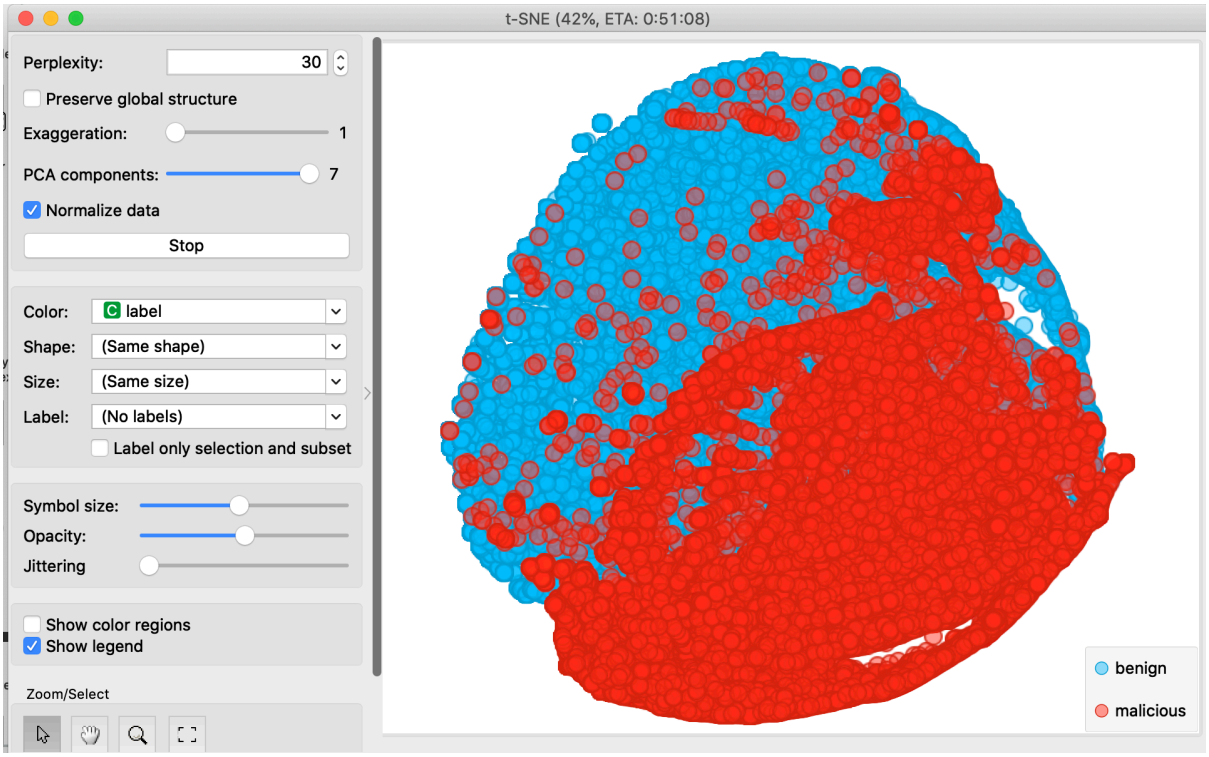
```
['http://accounts.servicelogin-service-mail.continue.https.mail.g00gle.com-en-us.co.uk/continue-https-maiiil.goooogle.com-
 'http://accounts.servicelogin-service-mail.continue.https.mail.g00gle.com-en-us.co.uk/continue-https-maiiil.goooogle.com-
 'http://loginprodx.att.net/commonLogin/igate_edam/controller.do?TAM_OP=login&amp;USERNAME=unauthenticated&amp;ERROR_CODE
 'http://loginprodx.att.net/commonLogin/igate_edam/controller.do?TAM_OP=login&USERNAME=unauthenticated&ERROR_CODE=0x00000
 'https://loginprodx.att.net/commonLogin/igate_edam/controller.do?TAM_OP=login&USERNAME=unauthenticated&ERROR_CODE=0x0000
 'https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https%3A%2F%2Fdocs.google.com%2Fdocument
 'https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https%3A%2F%2Fdrive.google.com%2Ffile%2F
 'https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https%3A%2F%2Fdocs.google.com%2Fa%2Fande
 'https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https%3A%2F%2Fdrive.google.com%2F%23&fol
 'https://login.yahoo.com:443/config/mail?.intl=au&.done=https%3A%2F%2Flogin.yahoo.com%3A443%2Fconfig%2Fmail%3F.intl%3Dau
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=658550133&amp;request_uri_path=/e/34442/user-ProtoLabsInc/2111w
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=658550133&amp;request_uri_path=/e/34442/protolabs/2111w8v/65855
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=658550133&amp;request_uri_path=/e/34442/ProtoLabs/2111w8q/65855
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=658550133&amp;request_uri_path=/e/34442/2018-12-06/2111w8n/6585
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=658550133&amp;request_uri_path=/e/34442/duck-html/2111w8g/65855
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=719333301&amp;request_uri_path=/e/54552/sow-edlbp001a-node-1564
 'https://href.li/?https://www.paypal.com/webapps/auth/protocol/openidconnect/v1/authorize?client_id=AbKEEezWscHjRlY3q3Jz
 'https://deref-mail.com/mail/client/ckGXDP5a6c0/dereferrer/?redirectUrl=https%3A%2F%2Fderef-mail.com%2Fmail%2Fclient%2Fs
 'https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https://drive.google.com/open/&followup=
 'http://inmyway.org/assets/auto/9951d69c1c958c84ca0f7a22a7a7d2a5/pdf.php?https://adobeid-na1.services.adobe.com/renga-id
 'http://inmyway.org/assets/auto/070792836652b10467d61c627a6fd17c/pdf.php?https://adobeid-na1.services.adobe.com/renga-id
 'https://go.pardot.com/tracker/httpsRedirect?pi_email_id=612117139&amp;request_uri_path=/e/62402/cute-overload-cpu-overl
 'https://accounts.google.com/ServiceLogin?service=CPanel&amp;passive=1209600&amp;cpbps=1&amp;continue=https%3A%2F%2Fadmi
```

在網址參數後帶其他有https的網站。再來看看變數與目標變數的correlation，https數量跟target有高達0.94的負相關程度，很大的決定是否為惡意連結的因素。

	result	ntld	url_len	special_count	digits	domain_len	domain_tokens	https_count
result	1.000000	0.134584	0.085057	0.027729	0.181187	−0.087717	−0.441141	−0.949651
ntld	0.134584	1.000000	0.015825	0.037103	−0.000503	0.152267	0.245705	−0.132510
url_len	0.085057	0.015825	1.000000	0.841937	0.734699	0.153325	0.060103	−0.054477
special_count	0.027729	0.037103	0.841937	1.000000	0.573125	0.075750	0.111617	−0.007221
digits	0.181187	−0.000503	0.734699	0.573125	1.000000	0.015708	−0.013448	−0.156394
domain_len	−0.087717	0.152267	0.153325	0.075750	0.015708	1.000000	0.498989	0.094410
domain_tokens	−0.441141	0.245705	0.060103	0.111617	−0.013448	0.498989	1.000000	0.431662
https_count	−0.949651	−0.132510	−0.054477	−0.007221	−0.156394	0.094410	0.431662	1.000000

VI. Data Visualization

原本我想用PCA進行降維把資料點畫在二維的平面上，但是因為我的變數本來就不多，只有兩個component的解釋變異不足夠表示這個資料集，所以就改用t-SNE將資料畫在平面上，就圖的結果來說是有明顯的分界區分好壞連結。



VII. Conclusion

Improvement in the Future

- More features
  - is shorteners?
  - Entropy
  - Length of query
  - Argument path
  - log length features
- A little bit imbalance
- Send requests to crawler more information

就這個資料集來說，應該是有挑過好壞連結的，因為有沒有加密SSL成為很判定是否為惡意連結的重要程度，相關性也是大得誇張，只要有做出是否有https就可以有很好的效果。未來改進的方向可能要做出更多特徵，例如 entropy、query長度、參數路徑、對現在的變數取log等等，或是去爬網站以取得更多資訊，讓好壞連結更能夠分別。這資料也稍微不平衡，可以做under sampling或是over sampling讓數據平衡一些來改善分類效果。