

Configuration Manual

MSc Internship Cyber Security

Dai Hoang Vu Student ID: x17165423

School of Computing National College of Ireland

Supervisor:

Ben Fletcher

National College of Ireland

MSc Project Submission Sheet



School of Computing

Student Name:	Dai Hoang Vu			
Student ID:	X17165423			
Programme:	Cyber Security		Year:	2019
Module:	Internship			
Supervisor:	Ben Fletcher			
Due Date:	12/12/2019			
Project Title: Word	Using Domain-Bas	ed on Machine Learning f	or Malwa	are Detection
Count:	4774	Page Count: 18		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: Dai Hoang Vu

Date: 12/12/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	х
Attach a Moodle submission receipt of the online project	x
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both	х
for your own reference and in case a project is lost or mislaid. It is not	
sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Using Domain-Based on Machine Learning for Malware Detection

Configuration Manual Submission

Dai Hoang Vu – X17165423

1. Install Tools

Step to Step for running the Machine Learning Algorithms to solves the problem about detecting Malicious Domain:

- Download and Install Anacoda[1] : <u>https://www.anaconda.com/distribution/</u>
- Find Cmd and activate tendor
- Download and Install PyCharm: <u>https://www.jetbrains.com/pycharm/download/#section=windows</u>
- Install Python 3.6
- In PyCharm, create new Project also go Setting, choose Project and click Existing interpreter, in here choose the folder which Anacoda was install and select python
- From the terminal of the Pycharm, you have to install some package like tensorflow, keras, skript,... and definitely install jupyterlab ... all package easy install by the command pip install jupyter-lab[2][3].
- After installing every package you need, from the terminal again, type : jupyter-lab. The Pycharm will create the host and which send you to the Jupyter lab to start the code.
- Copy all data set (Good and bad domain) which you downloaded before to the folder called data inside the PycharmProject.
- From there just follow every step which I show in these pictures because in the code, I mentioned already which code use for what missions. Or just open the Pycharm and start the project with Jupyter-lab you will see whole process without typing code yourself
- Source Code Download Link : <u>https://drive.google.com/file/d/1c-eXw1qs3XnyPSdjv76--</u> 7zhib7qjmXd/view

2. Start Coding

- Import Package : Import algorithms from the Scikit Learn library

8 + 3	🗋 📩 🕨 🔳 C' Code	v	Python 3 (
	Project : Using Domain-Based o	Machine Learning for Malware Detection	
	Dai Hoang Vu		
	Algorithms used :		
	- Random Forest		
	- Logistic Regression		
	- Naive Bayes		
	Data Gathering		
	Legit Domain : Alexa https://g	thub.com/mozilla/cipherscan/tree/master/toolm	
	Malicious Domain :		
	360 Lab Domain (over 1m domai	ns)	
	https://data.netlab.360.com/fe	<u>rds/dga.txt</u>	
	First Step : Set up		
	import numpy as np		
	import pandas as pd		
	import re		
	from publicsuffixlist import P	JblicSuffixList	
	import gc		
	import math		
	import collections		
	import matpiotiib.pypiot as pi	2	
	import seaborn as sns	anath topic topic colit	
	from sklearn matrics import as	nerv cran_cest_spin	
	from sklearn metrics import co	nusion matrix	
	from sklearn, ensemble import H	andomEorestClassifier	
	from sklearn, naive bayes impor	t GaussianNB	
	from sklearn.naive bayes impor	t MultinomialNB	
	from sklearn.naive bayes impor	BernoullinB	
	from sklearn.linear_model impo	Ht LogisticRegression	
	from sklearn.metrics import ro	curve, auc, roc_auc_score	
	from keras.preprocessing impor	t sequence	
	Anna hanna madala damant famo	-+1-1	

- Import Good and Bad Domain also Combine it all in One

Contit	realip	ynb	2										
8 +	*	00) - = (C Code	~		Python 3	0					
		RANDOM_	SEED = 1 tlib inline	-0									
	Using TensorFlow backend.												
		Put Ale	xa 1 milion	s domain									
	<pre>IB]: good_domain = pd.read_csv('data/top-1m-domain.csv', header=Nome, names=['Domain']) good_domain('D6A_Family'] = 'nome' good_domain('Type'] = 'nome' good_domain = good_domain[['D6A_Family','Domain','Type']] good_domain = dscribe(')</pre>												
	18]:		DGA_Family	Domain	Туре								
		count	1000000	1000000	1000000								
		unique	1	1000000	1								
		top	none	zuk-media.com	Normal								
		freq	1000000	1	1000000								
	<pre>[19]: bad_domain = pd.read_table('data/360_dgs.txt',namess['DGA_Family','Domain','Start_time','End_time']) bad_domain('type')='DGA' bad_domain('type')='DGA' bad_domain.to_csv('dats/360_dgs_domain.csv', index = False) bad_domain.describe()</pre>												
	19]:		DGA_Family	Do	main	Туре							
		count	1169720	116	59720 11	69720							
		unique	42	114	47770	1		*					

Untitled.ip	pynb	×				
+ %		► ■ C	Code ~		Pythor	13 0
	Attribut	eError: 'floa	it' object has no att	ribute 'split		
	Make all	Dommain in o	one file			
[23]:	bad_doma bad_doma all_doma all_doma	<pre>in = pd.conco in = bad_dom in = pd.conco in.describe()</pre>	at([bad_domain,bad1_c ain.drop_duplicates() at([bad_domain,good_c)	domain],axis=0) domain])		
[23]:	r.	OGA_Family	Domain T	ype		
	count	2183900	2183911 2183	911		
	unique	64	2183847	2		
	top	none lau	gqaoqixofl.ddns.net E	DGA		
	freq	1000000	2 1183	911		
	Save all	data to csv	file			
[24]:	all_doma all_doma	in_shuffle = in_shuffle.to	all_domain.sample(fr p_csv('data/all_domai	rac = 1, rando in.csv', index	n_state=RANDOM_SEED) = False)	
	all_doma	in_shuffle.h	ead()			
		DGA_Family	Domain	Туре		
	598985	none	ytsmovies.ga	Normal		
	669719	none	triger.com.pl	Normal		
	522313	none	fishintrepid.com	Normal		
	752250	banjori np	Iramentalistfanchonut.com	DGA		
	331668	none	bikepricesinnepal.com	Normal		
Untitled.ip	ynb	×				
+ %	331668	▶ ■ C	Code ~ bikepricesinnepal.com	Normal	Python	3 0
	This sect length me (CTS),Unc ratio of Now creat	tion will sho can(SLM), hav derscore Rati repeated cha te the datase	w how we create the e www Prefix(HwP), H o (UR), Contains IP racters in a subdoma t	dataset about as a Valid Top address(CIPA), in(RRC), The r	attributes for domain. It is include DGA name, lenght of domain(DNL), number of subdomain (No5),subdomain Level Domain (MVTLD), Contains Single-Character Subdomain (CSCS), Contains Top Level Domain as Subdomain Contains digit (contains digit), The ratio of vowel(vowel_ration), The ratio of digit(digit_ratio), The atio of consecutive digits(RCD), The entropy of subdomain(Entropy)	
[26]:	domain_wi	thFeatures =	all_domain_shuffle.	copy()		
[26].	domain_wi	GA Family	ead()	Type		
	598985	none	ytsmovies.ga	Normal		
	669719	none	triger.com.pl	Normal		
	522313	none	fishintrepid.com	Normal		
	752250	banjori npl	ramentalistfanchonut.com	DGA		
	331668	none	bikepricesinnepal.com	Normal		
	# Load Vo import sy	alid Top Leve rs	l Domains data			
	topLevel(with oper for]	Domain = [] n('data/1.txt Line in conte copLevelDomai	', 'r') as content: nt: n.append((line.strip	('\n')))		
	print(top	DLevelDomain)				
	['AAA', ' E', 'ACTO RBUS', 'A	AARP', 'ABAR R', 'AD', 'A IRFORCE', 'A	TH', 'ABB', 'ABBOTT' DAC', 'ADS', 'ADULT' IRTEL', 'AKDN', 'AL'	, 'ABBVIE', 'A , 'AE', 'AEG', , 'ALFAROMEO',	BC', 'ABLE', 'ABOGADO', 'ABUDHABI', 'AC', 'ACADEHY', 'ACCENTURE', 'ACCOUNTANT', 'ACCOUNTANTS', 'ACO', 'ACTIV 'AERO', AETNA', 'AF', 'AFANTLYCOPANN', 'AFL', 'AFRICA', 'AG', 'AGANHNI', 'AGENCY', 'AI', 'ATG', 'ATGO', 'A' 'AILBABA', 'ALTARY', 'ALLIFANZ', 'ALLIFATE', 'ALLY', 'AISACE', 'ALGION', 'AN', 'AMERICAMEPRESS', 'AMERICA	AI

- Set up code for each attribute. This set-up code helps the computer to recognize values, so that the computer can learn and make future assessments. For example, the Type variable, we will divide the good Domain with the value 0 and the bad domain variable with the value 1.

8 + %	; □ □ ▶ ■ C Code ∨
	Set up code for each Attributes
[28]:	<pre>osl = PublicSuffixList()</pre>
	def isnore/PS(domain):
	# Return the rest of domain after ignoring the Valid Public Suffixes:
	<pre>validPublicSuffix = `.' + psl.publicsuffix(domain) #f len(validPublicSuffix) < len(domain):</pre>
	# If it has VPS substring = domain[0::domain.index(validPublicSuffix)]
	<pre>elif len(validPublicSuffix) == len(domain):</pre>
	return 0 else:
	# If not subString = domain
	return subStrine
	eer typelo_largy(type); # Convert Type to Binary variable DGA = 1, Normal = 0
	if type == 'DGA': return 1
	else: return 0
	and densely locally (densely)
	er doması, engeri (doması); # Generate Domain Kome Length (DNL)
	return len(domain)
	def subdomains_number(domain): # Generate Number of Subdomains (NoS)
	subdomain = ImporeVPS(domain)
	return (subdomain.count(.) + 1)
	def subdomain_length_mean(domain): # enerate Subdomain Length Nean (SLM)
🖲 Untitled.ip	ynb X
8 + %	def subdomain length mean(domain):
	# enerate Subdomain Length Mean (SLM)
	<pre>subcomain = ignorevr>(commain) / subcomain.count('.')) / (subcomain.count('.') + 1)</pre>
	return result
	def has_waa_prefix(domain): # Generate has waa Perfix (http:/
	if domain.split('.')[0] == 'wnw':
	reurn i ale:
	neturn 0
	def has_hvltd(domain): # Generate Has a Valid Top Level Domain (HVTLD)
	<pre>if domain.split('.')[len(domain.split('.')) - 1].upper() in topLevelDomain:</pre>
	else:
	return 0
	def contains_single_character_subdomain(domain): # Generate Contains Single-Character Subdomain (CSSS)
	<pre>domain = ignoreWPS(domain) sts endits = domain solit(' ')</pre>
	minLength = len(str_split[0])
	<pre>ref i in runge(v)_imm(sr_split() - i); minlength = len(str_split([i]) if len(str_split[i]) < minlength else minlength</pre>
	if minlength == 1: return 1
	else: return 0
	def contains TLD subdomain(domain):
	# Generate Contains III as Subdomin (CTS)
	<pre>subdomain = ignoreVP>(domain) str_split = subdomain.split('.')</pre>
	<pre>for i in range(0, len(str_split) - 1): if str_split[i].upper() in toplevelDomain:</pre>
🖲 Untitled.ip)	ynb X
8 + %	The is manual lan(sta solit) - 1):
	if st_split[].upper() in topievelDomain:
	return 1 return 0
	def underscore_ratio(domain):
	# Generate Underscore Ratio (UR) on dataset
	<pre>result = subString.count('_') / (len(subString) - subString.count('.'))</pre>
	Peturn result
	def contains_IP_address(domain): # Generate Contains IP Address (CIPA) on datasetx
	<pre>splitSet = domain.split('.') for element in solitet:</pre>
	if(re.match("\d+", element)) == None:
	return 1
	def contains_digit(domain):
	Controlog Dialty
	subdomain = ignoreVPS(domain) for item im subdomain:
	if item.isdigit():
	return 1
	return 1 return 0
	return 0 def vouel_ratio(domain):
	return 0 def vowel_ratio(domsin): """ calculate Vowel Ratio
	return 1 return 0 def vomel_ratio(domain): ************************************
	return 1 return 0 def vomel_ratio(domsin): *** Calculate Vowel Ratio *** VONELS = set('aeioo') v_counter = 0
	<pre>reture 1 reture 0 def vowel_natio(domain): """ calculate Vowel Ratio """ VOELLS = set('aciou') v_counter = 0 a_counter = 0 ratio = 0 </pre>

🖲 Untitled.ipj	ynb ×		L.
B + X		Python 3 (5
	ratio = 0		
	subdomain = ignoreVPS(domain)		
	if item.isalpha():		
	a_counter+s1 f# item fm VUMFIS		
	v_counter+=1		
	if a_counter): natio = v counter/a counter		
	return ratio		
	def digit_ratio(domain):		
	taltulate ulgit Patio		
	d_counter = 0		Ł
	ratio = 0		Ł
	subdomain = ignoreVPS(domain) fer itee in subdomain:		
	if item.isalpha() or item.isdigit():		
	counter+=1.sidgit():		
	d_counter+=1		
	artics = d_counter/counter		
	return ratio		
	def prc_rrc(domain):		
	calculate the Ratio of Repeated Characters in a subdomain		
	subdomain = ignoreVP>(Gomain) subdomain = re.sub("[.]", ", ", subdomain)		
	char_nume0		
	<pre>repeater_instance d = collections.defaultdict(int)</pre>		
	fan 2 in lichtenheimsin):		٤.,
E Untitled.ip	pynb X		
8 + %	C □ ► ■ C Code ~	Python 3	0
	<pre>d = collections.defaultdict(int)</pre>		1
	for c in list(subdomain): d(c) += 1		
	for item in d:		
	<pre>cnar_mum +=1 if d[item]>1:</pre>		
	repeated_char_num +=1		
	retuo = repeateo_cnar_num/cnar_num return ratio		
	def ner ver(domain):		
	calculate the Ratio of Consecutive Consonants		
	<pre>VOWELS = set('aeiou')</pre>		
	conter = 0 con_counter=0		
	subdomain = ignoreVPS(domain)		1
	i = 0		
	if item.isalpha() and item not in VOWELS:		
	else:		
	if counter-i: cons counter+scounter		
	counter=0		
	1+=1 If fi=len(subdomain) and counter>1:		
	cons_counter+=counter		
	ratio = cons_counter/ien(subdomain) ration ratio		
	ere pr_rcu(usmaar):		
	calculate the ratio of consecutive digits		
	counter = 0		
Untitled.ip	ynb X		
H + %		Python 3	
	return ratio		
	def prc_rcd(domain):		
	calculate the ratio of conservive digits		
	calculate the Patio of consecutive digits		
	counter = 0 digit southernB		
	orgatiounce'+=> subdommin = ignoreVPS(domain)		
	<pre>fer item in subdomain: i = 0</pre>		
	if item.isdigit():		
	counter+=1 else:		
	if counter>1:		
	olfsr=Gomter=0		
	i+=1 if i=lensubdomain) and counter>1:		
	digit_counter+=counter		
	ratio = digit_courter/len(subdomain) reture ratio		
	ere projektivný konstaly		
	calculate the entropy of subdomain		
	:return: the value of entropy		
	subdomain = ignoreVPS(domain)		
	# get probability of chars in string		
	prov = [:tows(subound(c)) / len(subdomain) Tor c in dict.Tromkeys(list(subdomain))]		
	<pre># calculate the entropy entropy = - sum([p * math.log(p) / math.log(2.0) for p in prob])</pre>		
	return entropy		

+ %	ΚŌ	" ▶ ■	C Code	×.															Pytho
[29]: withFeatur	e #Crea def e d d d d d d d d d d e s.csv d d d d d d d d d d d d d d d d d d d	te each feat xtract_featu omain_withFe om	<pre>ure tres(): stures['DHL'] = stures['NoS'] = stures['NoS'] = stures['HVTLD'] stures['CSS'] stures['CSS'] stures['CSS'] stures['CTS'] = stures['CTA'] = stures['CTA'] = stures['RC'] = stures['RC'] = stures['RC'] = stures['RC'] = stures['RC']</pre>	domain_wi domain_wi domain_wi = domain_ = domain_wi domain_wit = domain_wi s_digit]= atio']= dc domain_wit domain_wit domain_wit domain_wit	ithFea ithFea withFea vithFea vithFea omain omain thFea thFea thFea thFea	atures[' atures[' atures[' eatures[atures[' atures[' atures[' withFee withFee tures['C tures['C feetures	Doma Doma Doma ['Dom Doma Domai 'Dom Feat ture Domai Comai Comai Comai	<pre>in'].ap in'].a in'</pre>	<pre>pply(lam pply(lam pply(lam pply(lam pply(lam pply(lam pply(lam pply(lam pply(lam ply(lam) ply(la</pre>	bda x: d bda x: s bda x: s bda x: d mbda x: d mbda x: d mbda x: d mbda x: d mbda x: d ply(lam ply(lam da x:pr da x:pr da x:pr da x:pr	domain subdon subdon has_ww : has_ contai ndersc contai lambda si bda x: c_rrc(c_rcc(c_rcc(x:prc	_length ains_nu ain_len w_prefi hvltd(x ins_ID_ ore_rat ins_IP_ x:cont vowel_r digit_r x)) x)) x)) x))	<pre>i(x)) imber(x)) igst_mean(x)) x(x)) igle_characte subdomain(x) iio(x)) address(x)) ains_digit(x atio(x)) atio(x)) </pre>	: :r_subdomain(x :))	;))				
[30]:	<pre>domain_withFeatures('Entropy']= domain_withFeatures['Domain'].apply(lambda x:prc_entropy(x)) # Generate features extract_features()</pre>																		
	# Cha domai	nge Type vir n_withFeatur	able from DGA a es['Type'] = do	<i>nd Normal</i> main_withf	to 1 eatur	and 0 res['Typ	e'].	apply(1	ambda ×	: typeT	o_Bina	ry(x))							
	Now h	ead to proce	ss the data																
	domai	n_withFeatur	es.head()		_														
	59898	DGA_Family		Domain Ismovies da	Туре	DNL N	1	SLM Hv	NP HVT	D CSCS	crs		PA contains_d	igit vowel_ratio	digit_ration	• RRC	0 444444	RCD	2 947703
	66971	9 none	ti	riger.com.pl	0	13	1	6.0	0	1 0	0 (0.0	0	0 0.333333	3 0.1	0 0.200000	0.3333333	0.0	2.251629
	52231	3 0000	fishir	ntrenid.com	0	16	1	12.0			0	0.0	0	0 000000		0.100000	0.416667	0.0	
								12.0	0	1 0	, ,	0.0	0	0 0.333333	5 0.2	0.100000	0.410007	0.0	3.188722
	TEADE	 bestelled 	and the second se			27		12.0	0	1 0	, ,	0.0	0	0 0.333333	3 0,	0.100000	0.410007	0.0	3.188722
	75225	o baniori	nolramentalistfan	chonut.com	1	27	1	23.0	0	1 0) 0	0.0	0	0 0.333333	3 D) 3 D/	0 0.266667	0.521739	0.0	3.675311
Untitled.i	75225	0 baniori ×	nolramentalistfan	chonut.com	1	27	1	23.0	0	1 () 0	0.0	0	0 0.304348	3 D, 3 D,	0 0.266667	0.521739	0.0	3.675311
Untitled.i	75225	baniori ×	C Code	chonut.com	1	27	1	23.0	0	1 0	0 0	0.0	0	0 0.304344	3 0,	0 0.266667	0.521739	0.0	3.188722 3.675311 Pytho
Untitled.i + 3	75225 ipynb K D : domai	o baniori × □ ► ■ n_withFeatur	c Code es.describe()	chonut.com	1	27	1	23.0	0	1 0) 0	0.0	0	0 0.30434	3 0, 3 0,	0.266667	0.521739	0.0	3.188722 3.675311 Pytho
Untitled.i + \$ [33]: [33]:	75225 ipynb K C domai	0 baniori × ⊡ ► ■ n_withFeatur Type	C Code es.describe() DNL	chonut.com	1	27 SLN	1	23.0 Hw	0 0	HVTLD	0	0.0	0 0 CTS	0 0.333333 0 0.304344 UR	3 0, 3 0, CIPA	0 0.266667	0.521739 jit vow	0.0 0.0 el_ratio	3.188722 3.675311 Pytho digit_ratio
Untitled.i + 3 [33]: [33]:	75225 ipynb K C domai	0 baniori × □ ► ■ n_withFeatur Type 2.183911e+06	C Code es.describe() DNL 2.183911e+06 2	Chonut.com No5 2.183911e+06	1	27 27 SLM 3911e+06	1 1 5 2.1	23.0 Hw 83911e+0	0 0 vP 06 2.183	1 0 1 0 HVTLD 911e+06	2.183	0.0 0.0 cscs	0 0 CTS 2.183911e+06	0 0.333333 0 0.304344 UR 2.183911e+06	CIPA 2183911.0	contains_dig 2.183911e+(0.521739 jit vow	0.0 0.0 el_ratio 11e+06	3.188722 3.675311 Pytho digit_ration 2.183911e+0
Untitled.i + 3 [33]: [33]:	75225 ipynb i domai count mean	0 baniori × □ ▶ ■ n_withFeatur Type 2.183911e+06 5.421059e-01	notramentalistfan C Code es.describe()	No5	1 i 2.18 i 1.38	27 27 SLM 3911e+00 3345e+01	1 5 2.1 1 5.4	12.0 23.0 Hw 83911e+0 403151e-0	vP 06 2.183 05 9.998	1 0 1 0 HVTLD 911e+06 1649e-01	2.183 ⁻ 6.776	0.0 0.0 CSCS 011e+06 833e-05	0 0 CTS 2.183911e+06 1.057735e-04	0 0.33333 0 0.304340 UR 2.183911e+06 5.711692e-07	CIPA 2183911.0 0.0	contains_dig 2.183911e+(1.305612e-(0.521739 it vow 06 2.1839 01 3.0606	0.0 0.0 el_ratio 11e+06 540e-01	3.188722 3.675311 Pytho digit_rati 2.183911e+0 3.405578e-0
Untitled.i + 3 [33]: [33]:	752250 ipynb K C : domai : count mean std	0 baniori × 1 ► ■ n_withFeatur 2,183911e+06 5,421059e-01 4,982241e-01	Dolramentalistfam C Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 2	No5	1 i 2.18 i 1.38 i 5.45	27 SLM 3911e+06 3345e+01 1104e+00	1 5 2.11 1 5.4 0 7.3	23.0 23.0 Hw 83911e+(803151e-(850416e-(vP 06 2.183 05 9.999 03 1.162	1 0 1 0 HVTLD 911e+06 1649e-01 1156e-02	2.183 6.776 8.231	0.0 0.0 cscs 011e+06 833e-05 875e-03	0 0 CTS 2.183911e+06 1.057735e-04 1.028408e-02	0 0.33333 0 0.30434(UR 2.183911e+06 5.711692e-07 2.088700e-04	CIPA 2183911.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(3.369199e-(0.521739 pit vow 06 2.1839 01 3.0606 01 1.3225	0.0 0.0 el_ratio 11e+06 540e-01 936e-01	3.188722 3.675311 Pytho digit_ratii 2.183911e+0 3.405578e-0 1.002824e-0
Untitled.i + 3 [33]: [33]:	752250 ipynb K C domai : count mean std min	baniori x x	Dolramentalistan C Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 2 2.000000e+00 1	No5 .183911e+06 .014205e+00 1.186975e-01	1 2.18 1.38 5.45 1.00	27 SLM 3911e+06 3345e+01 1104e+00 0000e+00	1 5 2.11 1 5.4 0 7.3 0 0.0	Hw 83911e+0 803151e-0 80000e+0	vP 06 2.183 05 9.999 03 1.162 00 0.000	1 0 1 0 HVTLD 911e+06 1649e-01 156e-02 000e+00	2.183 6.776 8.231 0.000	0.0 0.0 CSCS 011e+06 833e-05 875e-03 000e+00	0 0 CTS 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00	UR 2.183911e+06 5.711692e-07 2.088700e-04 0.00000e+00	CIPA 2183911.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(3.369199e-(0.000000e+(0.521739 0.521739 iit vow 06 2.1839 01 3.0600 01 1.3225 00 0.0000	0.0 0.0 el_ratio 11e+06 540e-01 936e-01 00e+00	3.188722 3.675311 Pythe digit_rati. 2.183911e+0 3.405578e-0 1.002824e-0 0.000000e+0
Untitled.i	752250 ipynb K C domai count mean std min 25%	baniori x x	Diramentalistram C Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 2.00000e+00 1 1.40000e+01 1	No5 183911e+06 014205e+00 1.186975e-01 0.000000e+00	1 2.18 1.38 5.45 1.00 1.00	27 SLM 3911e+06 3345e+01 1104e+00 0000e+00 0000e+01	1 5 2.11 1 5.4 0 7.3 0 0.0	Hw 23.0 Hw 83911e+(403151e-(00000e+(00000e+(vP 06 2.183 05 9.999 03 1.162 00 0.000 00 1.000	1 0 1 0 HVTLD 911e+06 1649e-01 156e-02 000e+00 000e+00	2.183 6.776 8.231 0.000	0.0 0.0 CSCS 211e+06 833e-05 875e-03 200e+00 200e+00	0 0 CTS 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00	UR 2.183911e+06 5.711692e-07 2.088700e-04 0.00000e+00 0.00000e+00	CIPA 2183911.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(3.369199e-(0.000000e+(0.000000e+(it vow 0.521739 it vow 06 2.1839 01 3.0606 01 1.3225 00 0.0000 00 2.1425	0.0 0.0 el_ratio 11e+06 540e-01 936e-01 00e+00 857e-01	3.188722 3.675311 Pytho digit_rati 2.183911e+0 3.405578e-0 1.002824e-0 0.000000e+0 0.000000e+0
Untitled.i + 3 [33]: [33]:	752250 ipynb K C domai count mean std min 25% 50%	baniori baniori x baniori x x x x x x x	Diramentalistran C Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 2.00000e+00 1 1.40000e+01 1	No5 1.183911e+06 0.014205e+00 1.186975e-01 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00	1 2.18 1.38 1.38 1.00 1.00 1.00	27 SLM 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0.0	Hw 83911e+0 803151e-0 803151e-0 00000e+0 00000e+0 00000e+0	VP 06 2.183 05 9.999 03 1.162 00 0.000 00 1.000 00 1.000	HVTLD 911e+06 8649e-01 8156e-02 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000	0.0 0.0 CSCS 211e+06 833e-05 875e-03 200e+00 200e+00 200e+00	0 0 CTS 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.00000e+00	UR 2.183911e-06 5.711692e-07 2.088700e-04 0.000000e+00 0.000000e+00	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(3.369199e-(0.000000e+(0.000000e+(0.000000e+)	iit vow 0.521739 0.521739 iit vow 06 2.1839 01 3.0606 01 1.3225 00 0.0000 02 2.1426 03 3.1816	0.0 0.0 el_ratio 11e+06 540e-01 936e-01 00e+00 357e-01 318e-01	3.188722 3.675311 Pytho 2.183911e+00 3.405578e-00 1.002824e-0 0.000000e+00 0.000000e+00
Untitled.i	752250 ipynb K C domai count mean std min 25% 50% 75%	baniori baniori baniori baniori clisted and a second and and a second and a	Noiramentalistian C Code es.describe() DNL DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 2.000000e+00 1 1.400000e+01 1 2.100000e+01 1 2.00000e+01 1	No5 183911e-06 0.014205e-00 1.186975e-01 0.00000e+00 0.000000e+00 0.00000e+00 0.00000e+00 0.0000000000	i 2.18 i 2.18 i 5.45 i 1.00 i 1.00 i 1.40 i 1.80	27 SLN 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0.0 1 0.0	Hw 83911e+0 803151e-0 80000e+0 00000e+0 00000e+0 00000e+0 00000e+0	vP 06 2.183 05 9.994 03 1.162 00 0.000 00 1.000 00 1.000	HVTLD 911e+06 1649e-01 156e-02 000e+00 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000 0.000	0.0 0.0 CSCS 011e+06 833e-05 875e-03 000e+00 000e+00 000e+00 000e+00	0 0 CTS 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00	0 0.33434 0 0.304344 2.183911e+06 5.711692e-07 2.088700e-04 0.000000e+00 0.000000e+00 0.000000e+00	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(3.369199e-(0.000000e+(0.000000e+(0.000000e+(0.000000e+(it vow 6 2.1839 1 3.0606 0 0.0000 0 2.1426 0 3.1816 0 4.0000	0.0 0.0 el_ratio 11e+06 540e-01 036e-01 00e+00 357e-01 318e-01 000e-01	3.188/22 3.675311 Pytho 2.183911e-0 3.405578e-0 1.002824e-0 0.000000e+0 0.000000e+0 0.000000e+0
Untitled.i + 3 [33]: [33]:	752250 ipynb K D : domai : count mean std min 25% 50% 75% max	banjori ► ■ n_withFeatur Type 2.183911e+06 5.421059e-01 4.982241e-01 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00	noiramentalistran C Code es.describe() ONL DNL 2.183911er06 2 1.796130er01 1 5.526419er00 1 2.000000er00 1 1.400000er01 1 1.900000er01 1 2.100000er01 1 2.100000er01 1 2.10000er01 1	No5 183911e-06 0.014205e-00 1.186975e-01 0.00000e+00 0.0000000000	1 ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	27 SLM 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 5 5.10 1 5.4 0 7.3 0 0.0 1 0.0 1 0.0 1 0.0	Hw 23.0 Hw 83911e+(83	vP 06 2.183 05 9.999 03 1.162 00 0.000 00 1.000 00 1.000 00 1.000 00 1.000	HVTLD 911e+06 1649e-01 2156e-02 000e+00 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000 0.000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	0 0 crs 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00	0 0.33433 0 0.304340 UR 2.183911e+06 5.711692e-07 2.088700e+04 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.388632e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e+(3.369199e+(0.000000e+(0.000000e+(0.000000e+(1.000000e+(1.000000e+(iii vow 0.521739 0.521739 iiii vow 06 2.1839 01 3.0600 01 1.3225 00 0.0000 02 2.1426 03 3.1816 00 4.0000 01 1.0000	0.0 0.0 0.0 el_ratio 11e+06 540e-01 936e-01 936e-01 9357e-01 818e-01 900e-01 900e+00	3.188/22 3.675311 Pythc 2.183911e+0 3.405578e-0 1.002824e-0 0.000000e+0 0.000000e+0 0.000000e+0 0.000000e+0
Untitled.i + 3 [33]: [33]:	752250 ipynb K C count mean std min 25% 50% 75% max	baniori compared to the second seco	notramentalistram Code A es.describe() DNL DNL 2.183911e+06 2.00000e+00 1.796130e+01 1 5.526419e+00 1 2.000000e+00 1 1.400000e+01 1 1.400000e+01 1 2.10000e+01 1 7.300000e+01 1 7.30000e+01 4	No5 No5 183911e-06 0.014205e-00 1.186975e-01 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00	1 2.18 1.2.18 1.38 1.00 1.00 1.00 1.40 0.1.40	27 SLM 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 9 7.3 9 0.0 1 0.0 1 0.0 1 0.0	Hw 83911e+(83915e-0 850416e-0 00000e+0 00000e+0 00000e+0 00000e+0	vP 06 2.183 05 9.994 03 1.162 00 0.000 00 1.000 00 1.000 00 1.000	HVTLD 911e+06 1649e-01 1156e-02 000e+00 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000 0.000 1.000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	0 0 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00	UR 2.183911e+06 5.711692e-07 2.088700e-04 0.000000e+00 0.000000e+00 1.538462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e+(3.369199e+(0.000000e+(0.000000e+(1.000000e+(1.000000e+(it vow 0.521739 0.521739 it vow 06 2.1839 01 3.0606 01 1.3225 00 0.0000 02 2.1426 03 3.1816 00 4.0000 00 1.0000	0.0 0.0 0.0 11e+06 540e-01 035e-01 00e+00 357e-01 318e-01 000e-01 000e+00	3.188/22 3.675311 Pytho 2.183911e+0 3.405578e-0 1.002824e-0 0.000000e+0 0.000000e+0 0.000000e+0 1.000000e+0
Untitled.i	752250 ipynb K C count mean std min 25% 50% 75% max 4	baniori b	noiramentalistran C* Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 1.400000e+01 1 2.100000e+01 1 2.100000e+01 1 7.30000e+01 4	No5 .183911e-06 .014205e-00 1.186975e-01 .000000e-00 .000000e-00 .000000e+00	1 2.18 1.2.18 1.38 1.00 1.00 1.00 1.40 0.1.80 0.30	27 SLM 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 9 7.3 9 0.0 1 0.0 1 0.0 1 0.0	Hw 83911e+(403151e-0 00000e+(00000e+(00000e+(00000e+(vP 06 2.183 05 9.994 03 1.162 00 0.000 00 1.000 00 1.000 00 1.000	HVTLD 911e+06 1649e-01 1156e-02 000e+00 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000 0.000 1.000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	0 0 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00	UR 2.183911e+06 5.711692e-07 2.083700e+00 0.000000e+00 0.000000e+00 1.538462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e+(3.36919e+(0.00000e+(0.000000e+(1.000000e+(it vow 0.521739 0.521739 it vow 06 2.1839 01 3.060e 01 1.3225 00 0.0000 02 2.1428 00 3.1816 00 4.0000 01 1.0000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	3.188/22 3.675311 Pytho 2.183911e+0 3.405578e+0 1.002824e+0 0.000000e+0 0.000000e+0 0.000000e+0 1.000000e+0
Untitled.i + 3 [33]: [33]: [33]:	752250 ipynb K C : domai : count mean std min 25% 50% 75% max : : : : : : : : : : : : :	baniori baniori c baniori c baniori c c baniori c	noiramentalistian C Code es.describe() DNL 2.183911e+06 1.796130e+01 2.000000e+00 1.400000e+01 1.900000e+01 7.30000e+01 4 es.to_csv('data	No5 1.183911e+06 0.014205e+00 1.186975e-01 1.000000e+00 0.0000000e+00 0.0000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.0000000000 0.0000000000000000000	1 i 2.18 i 2.18 i 1.38 i 5.45 i 1.00 i 1	27 SLM 3911e+06 3345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 9 7.3 9 0.0 1 0.0 1 0.0 1 0.0 1 0.0 1 1.0	Hw 83911e+(403151e-(403151e-(00000e+(00000e+(00000e+(100000e+(vP 06 2.183 05 9.999 03 1.162 00 0.000 00 1.000 00 1.000 00 1.000 00 1.000 00 1.000 00 1.000	HVTLD 911e+06 96649e-01 1156e-02 000e+00 000e+00 000e+00	2.183 6.776 8.231 0.000 0.000 1.000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	0 0 CTS 2.183911e+06 1.057735e-0 0.00000e+00 0.00000e+00 1.00000e+00	UR 2.183911-06 5.7.11692e-07 2.088700e-04 0.000000e+00 0.000000e+00 1.538462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	containa_dig 2.183911e+ 1.305612e+ 0.000000e+(0.000000e+(1.000000e+(it vow 0.521739 0.521739 it vow 06 2.1839 01 3.0606 01 1.3225 00 0.0000 02 2.1426 00 3.1816 00 1.0000 01 1.0000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	3.189/22 3.675311 Pythe 2.183911e+0 3.405578e-0 1.002824e-0 0.000000e+0 0.000000e+0 1.000000e+0
Untitled.) + 3 [33]: [33]: [33]: [33]: [35]:	752259 ipynb K count mean std min 25% 50% 75% max 4 50% 75% max 4 1 <i>f</i> source <i>f</i>	baniori x x x ype 2.183911e+06 5.421059e-01 4.922241e-01 0.000000e+00 1.000000e+00 1.00000e+00 1.0000000000 1.000000000000000000000	noiramentalistifan C* Code C* Code C* Code C* Code C* Code DNL DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 2 2.00000e+01 1 1.400000e+01 1 7.300000e+01 1 7.300000e+01 4 es.to_csv('data gsin es = pd.read_css	No5 183911e-06 .014205e-00 1.186975e-01 .000000e+00 .00000e+00 .000000e+00 .000000e+00 .0000000000000000000000000000000000	i 2.18 i 2.18 i 1.38 i 5.45 i 1.00 i 1.00 i 1.40 i 1.80 i 6.30 ithFe.	27 SLM 3911e+00 33345e+01 1104e+00 0000e+01 0000e+01 0000e+01 0000e+01 0000e+01 0000e+01	1 5 2.11 1 5.4 0 0.0 1 0.0 1 0.0 1 0.0 1 1.0 5 5 5 7.3 1 0.0 1 0.0	Hww 83911e+(0 003151e-(0 00000e+(0 000000e+(0 000000e+(0 000000e+(0 000000e+(0 0000000000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	HVTLD 911e+06 6649e-01 1156e-02 0000e+00 0000e+00	2.183 6.776 8.231 0.000 0.000 1.000	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	CTS 2.183911e+06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00	0 0.33333 0 0.004340 UR 2.113911+e06 5.711692e-07 2.083700e-04 0.000000e-00 0.000000e-00 0.000000e+00 1.5318462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e-(0.00000e+(0.000000e+(1.000000e+(it vow 0521739	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	3.189722 3.675311 Pytho 2.183911e+0 3.405578e-0 1.002024e-0 0.000000e+0 0.000000e+0 1.000000e+0
Untitled.il + 3 [33]: [33]: [33]: [33]: [33]: [33]: [34]: [35]: [36]:	752259 ipynb K Comai count mean std min 25% 75% max 75% count ean std count mean std min 25% 75% count ean std count ean sta sta count ean sta sta sta sta sta sta sta sta	banion X X A v	noiramentalistian C Code C Code DNL DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 2.00000e+01 1 1.400000e+01 1 2.10000e+01 1 7.30000e+01 1 es.to_csv(*data gpin es.thead()	Nos N	i 2.18 i 2.18 i 3.45 i 1.00 i 1.00 i 1.40 i 1.80 i	27 SLM 3911e+06 33345e+01 1104e+00 00000e+01 00000e+01 00000e+01 atures.c	1 5 2.11 1 5.4 9 7.3 9 0.0 1 0.0 1 0.0 1 0.0 1 0.0 1 1.0	Hww 83911e+C 000000+C 000000+C 000000+C 000000+C indexs \$.csv')	<pre> 0</pre>	HVTLD 911e+06 6649e-01 1156e-02 0000e+00 0000e+00 0000e+00	2.183 6.776 8.231 0.0000 0.0000 0.0000	0.0 0.0 0.0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	CTS 2.183911+e-06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00	0 0.33333 0 0.004340 2.113911+e06 5.711692e-07 2.08700e-04 0.000000e+00 0.000000e+00 0.000000e+00 1.538462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911+(1.305612e-(0.000000e+(0.000000e+(1.000000e+(it vow jit vow jit vow jit solo	0.0 0.0 11e+06 540e-01 00e+00 357e-01 318e-01 00e+00	3.189722 3.675311 Pytho 2.183911e+0 3.405578e-0 0.000000e+0 0.000000e+0 0.000000e+0
Untitled.i + 3 [33]: [33]: [34]: [36]: [36]: [36]:	752259 ripynb K Domai count mean std min 25% 50% 75% domai # Loo domai # Loo domai Domai	baniori x x yee 2.183911e+06 5.421059e+01 4.982241e-01 0.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 the data n_xithFeatur d the data an_withFeatur	noiramentalistian C Code C Code DNL DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 1.00000e+01 1 1.400000e+01 1 2.100000e+01 1 7.300000e+01 1 7.300000e+01 1 es.to_csv(*data gpin es = pd.read_cs es.head() Do	No5 1.183911e-06 0.014205e-00 1.186975e-01 1.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.000000e-00 0.00000e-00 0.00000e-00 0.00000e-00 0.00000e-00 0.00000e-00 0.00000e-00 0.00000e-00 0.0000000e-00 0.000000e-00 0.00000000 0.000000000 0.000000000	<pre>i 2.18 i 2.18 i 2.18 i 3.45 i 1.00 i 1.00 i 1.40 i 1.80 i 6.30 ithFe. omain DNL</pre>	27 SLM 3911e-00 0000e+01 0000e+01 0000e+01 0000e+01 0000e+01 NoS 3	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0.0 1 0.0 1 0.0 1 1.0 5 1 1.0 5 1 1.0	HwwP H	vP 06 2.183 05 9.994 03 1.16; 00 0.000 0 0 0.000 0 0 0 0 0 0 0 0 0 0 0	HVTLD 911e+06 6649e-01 1156e-02 0000e+00 0000e+00 0000e+00	2.183 6.776 8.231 0.000 0.000 1.000 1.000	CIPA ct	0 0 CTS 2.183911+e-06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00 0.00000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000000000000000000000000000000	0 0.33333 0 0.00434 UR 2.113911+e06 5.711692e-07 2.08700e-04 0.000000e-00 0.000000e-00 0.000000e-00 1.538462e-01 1.538462e-01	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+(1.305612e+(0.000000e+(0.000000e+(1.000000e+(it vow 0.521739 it vow 0.521739 it automotion 1.3225 0.0000 0.13225 0.0000 0.14225 0.14225 0.14205 0.142555 0.142555 0.142555 0.142555 0.142555 0.1425555 0.142555555	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	3.189722 3.675311 Pytho 2.183911e-0 3.405578e-0 0.000000e-0 0.000000e-0 0.000000e-0 1.000000e-0 1.000000e-0
Untiled, (33): (3): (752250 ipynb K C domai count mean std min 25% 50% 75% max 4 count i <i>t</i> <i>t</i> <i>t</i> <i>t</i> <i>t</i> <i>t</i> <i>t</i> <i>t</i>	baniori x x x yee 2.183911e+06 5.421059e-01 4.922241e-01 0.00000e+00 0.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 thousand the data n_vithFeatur n_vithFeatur n,vithFeatur in, isthFeatur i	noiramentalistian C* Code c Code c DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 2 2.00000e+00 1 1.00000e+01 1 2.100000e+01 1 7.30000e+01 4 es.to_ccv('data gpin es.to_ccv('data es.head() De	Nos 183911e+06 0.014205e+00 1.186975e-01 0.000000e+00 0.00000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.00000e+00 0.0000e+00 0.0000e+00	1 2 138 3 2.18 3 2.	SLM SLM 3911e+06 3335e+01 1104e+00 00000e+0 00000e+01 00000 00000e+01 000000	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0.0 1 0.0 1 0.0 1 0.0 1 1.0 1 1	Hwe H 0	vP 06 2.183 05 9.994 00 0.000 00 1.000 00 1.000 00 1.000 00 1.000 1.000 00 1.000 00 000 0000 0	HVTLD 911e+06 4649e-01 1156e-02 0000e+00 0000e+00 0000e+00 0000e+00 0000e+00 0000e+00 0000e+00	2.183 6.776 8.231 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.00000 0.000000	0.00 0.0 CSCS 111e+06 833e-05 875e-03 0000e+00 000000	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0.33333 0 0.00434 UR 2.183911+e06 5.711692e-07 2.085700e-04 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000000000e+00 0.000000000000000000000000000000000	CIPA 2183911.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	contains_dig 2.183911e+1 1.305612e+1 3.369199e+0 0.000000e+1 0.000000e+1 1.000000e+1 1.000000e+1 2.000000e+1 1.000000e+1 2.000000e+1 1.000000e+1 1.000000e+1 2.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000000+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.0000000e+1 1.0000000e+1 1.00000000+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.000000e+1 1.000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.000000e+1 1.0000000e+1 1.000000e+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.0000000e+1 1.00000000000000000000000000000000000	it vow jt vow jt vow jt jt jt	0.0 0.0 11e+06 540e-01 336e-01 00e+00 00e+00 00e+00	3.189722 3.675311 Pythc digit_ratic 2.183911=0 3.405578=0 1.002824=0 0.000000=0 0.000000=0 1.000000=0 1.000000=0
Untiled, (33): (33): (33): (33): (33): (34): (34): (34): (34): (34): (34): (34): (34): (34): (35): (36):	752255 ipynb K C count mean std min 25% 50% 75% max 4 Count t mean std count t mean std count t count t mean std count t count	0 baniori × 1 vithFeatur 2.183911e+06 5.421059e-01 4.982241e-01 0.00000e+00 1.000000e+00 1.000000e+00 1.000000000	noiramentalistian C Code cs.describe() ONL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 1 2.00000e+00 1 1.400000e+01 1 2.100000e+01 1 2.100000e+01 1 7.300000e+01 1 es.to_csv('data gpin es.tead_cs es.head() Do ytsmovi triger.or	Koss Kess Kes Kess Kess Kess	1 2 138 3 2.18 3 2.	27 SLM SLM 39911+-00 00000+001 00000+001 000000+01 000000+01 000000+01 000000+01 000000+01 000000+01 00000+00000000+0000000000	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0	Hww H 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<pre>vP vP vP vF vF vF vF vF vF vF vF vF v v v v</pre>	HVTLD 1 C HVTLD 911e+06 6649e-01 1156e-02 0000e+00 00000e+00 00000 00	2.1839 6.776 8.231 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	0.0 0.0 CSCS 011e+06 833e-05 875e-03 000e+00 0000e+0000000000	0 0 CTS 2.183911e-06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 1.000000e+00 1.000000e+00 0 1.000000e+00	UR 2.183911e-06 5.7.11692e-07 2.082700e-04 0.000000e-00 0.000000e-00 1.538462e-01 1.538462e-01	CIPA CIPA CIPA CIPA CIPA 00 00 00 00 00 00 00 00 00 0	contains_dig 2.183911e+ 1.305612e- 0.000000e+ 0.000000e+ 1.000000e+ 1.000000e+ 1.000000e+ 2.5000 0.44 25000 0.44	it vow jit vow jit vow jit so	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	3.189722 3.675311 Pythe 2.183911e-00 3.405578-00 1.002824e-01 0.000000e+00 0.000000e+00 1.000000e+00 1.000000e+00
Untiled, [33]: [33]: [33]: [33]: [33]: [33]: [34]: [36]: [36]:	73229 ipynb K C count mean std min 25% 75% max 4 <i>d</i> omai <i>d</i> omai <i>d d</i> omai <i>d d</i> omai <i>d d</i> omai <i>d d d d d d d d d d</i>	banion x banion x x x x x x x x x x x x x	noiramentalistfan C Code es.describe() DNL 2.183911e+06 2 1.796130e+01 1 5.526419e+00 2 2.00000e+00 1 1.90000e+01 1 1.90000e+01 1 2.10000e+01 1 2.10000e+01 1 2.10000e+01 1 2.10000e+01 1 2.10000e+01 1 es.to_csv('data gsin es.to_csv('data	Nos Nos 1183911e+06 0014205e+00 1.186975e-01 1.000000e+00 0000000e+00 0000000e+00 0000000e+00 0000000e+00 0000000e+00 0000000e+00 000000e+00 00000e+00 0000e+00 00000e+00 00000e+00 00000e+00 00000e+00 000	1 2 2.18 3 2.18 3 1.39 1 5.45 3 1.00 9 1.40 9 1	SLM SLM 39114+00 00000+00 00000+01 00000+0000+	1 5 2.11 1 5.4 0 7.3 0 0.0 1 0	Hun B3911e+C 000000e+C 000000e+C 000000e+C 000000e+C 000000e+C Hundex= Hundex= 0 0 0 0	vP 0 0 0 0 0 0 0 0 0 0 0 0 0	HVTLD 911e+06 6649e-01 1156e-02 0000e+00 0000e+00 0000e+00 0000e+00 0000e+00 0000e+00 0 0000e+00 0 0 0 0 0 0 0 0 0 0 0 0	2.183 6.776 8.231 0.000 0.000 0.000 1.000 1.000 1.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	CSCS CSCS 111e+06 8338-05 875e-03 000e+00	0 0 CTS 2.183911e-06 1.057735e-04 0.00000e+00 0.00000e+00 0.00000e+00 1.00000e+00 0.000000e+00 0.000000000000000000000000000000000	UR 2.183911-06 5.7.11692e-07 2.08700e-04 0.00000e-00 0.00000e-00 0.00000e-00 1.538462e-01	i 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	contains_dig 2.183911e+(1.305612e-(0.000000e+(0.000000e+(0.000000e+(1.0000000e+(1.000000e+(1.0000000e+(1.000000000000000000000000000000000000	it vow	el, ratio 11e+06 540e-01 136e-01 00e+00 00e+00 00e+00 00e+00 Entre 2.947 2.2511 3.188	3.189722 3.675311 Pythe 2.183911e-0 3.405578-0 0.000000e-0 0.000000e-0 0.000000e-0 1.000000e-0 1.000000e-0 1.000000e-0 9 9999 703 529 722
Untiled, (33): (33): (33): (33): (33): (33): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (34): (35):	73229 ipynb K D domai std min std min 25% max 4 <i>V</i> Loa domai <i>V</i> Loa domai <i>V</i> Loa domai <i>V</i> Loa domai <i>V</i> Loa <i>V</i> Lo	baniori x x x yee 183911e+06 5.421059e-01 4.922241e-01 0.00000e+00 0.00000e+00 1.00000e+00 1.0000e+00 1.00000e+00 1.0000e+00 1.0000e+00 1.0000e+00 1.0	noiramentalistran C Code C Code S describe() DNL 2.183911e+06 2.1796130e+01 1 S.S.26419e+00 2 2.00000e+00 1 1.400000e+01 1 2.100000e+01 1 2.100000e+01 1 7.300000e+01 4 gsin es.to_csv('data gsin es.thead() De ytsmovi triget.or fishinterpio amentalistranchonum triget.or	Nos N	1 2 2.18 3 2.18 3 1.38 3 1.38 3 1.40 3 1	SLM SLM SLM SLM SLM SLM SLM SLM	1 5 2.11 1 5.4 9 7.3 9 0.0 1 0	Hw HwP H 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	HVTLD 911e+06 6649e-01 1156e-02 000e+00 00000000	2.183 6.776 8.231 0.000 0.000 1.000 1.000 5. UR 9. 0.0 9. 0.0 9.0	0.0 0.0 CSCS 111e+06 833e-05 875e-03 000e+000 000e+00 0000 000e+00	0 0 CTS 2.183911e-06 1.057735e-04 1.028408e-02 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.000000000000000000000000000000000	0 0.33333 0 0.00434 UR 2.183911+e06 5.711692e-07 2.08770e-04 0.000000+00 0.0000000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.00000+00 0.0000+0000+	Image: 1 0.0 Image: 2 2183911.0 Image: 0 0.0 Image: 0 0.0	contains_dig 2.183911e+1 3.369199e+(0.00000e+(0.00000e+(0.00000e+(1.0000000e+(1.0000000e+(1.0000000e+(1.00000000000e+(1.0000000e+(1.000000000000000000000000000000000000	it vow jit vow jit vow jit a jit	el_ratio 11e+06 540e-01 135e-01 00e+00 00e+00 00e+00 00e+00 1 2.2511 3.188 3.675	3.189722 3.675311 Pythy digit_reti 2.183911=0 3.405578=0 0.000000=0 0.000000=0 0.000000=0 0.000000=0 1.000000=0 1.000000=0 1.000000=0 1.000000=0 1.000000=1 1.0000000=1 1.000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.0000000=1 1.00000000=1 1.000000000=1 1.0000000000

- Create Feature for Domain Attributes: Remove some attribute not important for ML algorithms

- Create the table: the values with blue color on the table show little correlation between the two variables, we will remove these variables to reduce redundancy for the algorithms we apply next.



Prepare variable for Training Dataset and Testing Dataset[4]

E Ur	title	d.ipy	nb		×																
а	+	Ж	Ō	D •		с	Code	~													Python 3
			75%	1.00000	0e+00	2.1	00000e+01	1.00	0000e+00	1.80	0000e+01	0.0000006	e+00	0.0	0.000000e+00	4.000000e-01	0.000000e+00	3.636364e-01	5.000000e-01	0.000000e+00	3.577820e+00
			max	1.00000	0e+00	7.3	00000e+01	4.00	0000e+00	6.30	0000e+01	1.000000	e+00	0.0	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	9.583333e-01	9.000000e-01	4.954196e+00
	[4	1]:	domai	n_withF	eatur	es_f	ixed.ism	ull()	.sum()												
	[4	6]:	Type DNL NoS SLM CTS CIPA conta vowel digit RRC RCC RCD Entrop	ins_dig _ratio _ratio Py	it	000000000000000000000000000000000000000															
			Prepa	ir for	train	ing	and test	ing d	ataset												
	[4		<pre># Show independent variables and dependent variables attributes = domain_withFeatures_fixed.drop('type', axis=1) observed = domain_withFeatures_fixed['type'] attributes.hape, observed.shape</pre>																		
	[4		((218	3911, 1	2), (3	2183	911,))														
	[4		# Spl train train	it the _X, tes _X.shap	datas t_X, e, te	et i trai	into train in_y, test Lishape, f	ning (t_y = train,	dotoset train_i _y.shap	and i test_s	testing split(at st_y.sha	dataset tributes, pe	observed	l, tes	st_size = 0.2	5, random_st	ate = RANDOM	_SEED)			
			((163	7933. 1	2). (5459	78, 12).	(163)	7933.).	(5459	978.))										

Naïve bayes algorithms[5] : This algorithm is processed in less than 1s.

-

•	+ %	C Python Python	3 0
		Naive Bayes Algorithms	
	[47]:	# Use Gaussian Naive Bayse to build a model gob = GaussianNM()	
		gno.iit(train_s, train_y)	
		# Get the prediction	
editot		train_mnb_pred = gnb.predict(train_X)	
COLON		test_gnb_pred = gnb.predict(test_X)	
	[48]:	# Caluate the accuracy	
		<pre>score_gnb_train = round(accuracy_score(train_y, train_gnb_pred) * 100, 2)</pre>	
		<pre>score_gnb_test = round(accuracy_score(test_y, test_gnb_pred) * 100, 2)</pre>	
		print("Accuracy of usussian Naive Dayes on training dataset: ", score_gno_train) print("Accuracy of daussian Naive Bayes on test dataset: ", score_gno_train)	
		and the second	
		Accuracy or usenssian naive payes on training dataset: 00.52 Accuracy of Gaussian Naive Bayes on test dataset: 66.6	
	[49]:	# Use Nultinomial Naive Bayes Nodel	
		mb = malinotaino(appmav.s/	
		<pre>test_mnb_pred = mnb.predict(test_X)</pre>	
		<pre>train_mnb_pred = mnb.predict(train_X)</pre>	
		# Calculate the accument	
		s core mb train = round(accuracy score(train y, train mb pred) * 100, 2)	
		<pre>score_mnb_test = round(accuracy_score(test_y, test_mnb_pred) * 100, 2)</pre>	
		print("Accuracy of Multinomial Naive Bayes on training dataset: ", score_mnb_train)	
		print("Accuracy of Multinomial Naive Bayes on test dataset: ", score_mnb_test)	
		Accuracy of Multinomial Naive Bayes on training dataset: 79.94	
		Accuracy of Multinomial Naive Bayes on test dataset: 79.94	
		# Use Bernoulli Naive Bayes Model	
		bnb = BernoullinB(alpha=0.9)	
		DOD-IIS(Iran_X, tran_y)	

- Logistic Regression Algorithms[6]: This algorithm is processed in 30 seconds.



Random Forest Algorithms[7]: This algorithm is processed in 5 minutes and 15 seconds.



Thanks for watching my tutorial

3. References :

[1] Docs.anaconda.com. (2020). User guide — Anaconda documentation. [online] Available at: https://docs.anaconda.com/anaconda/user-guide/ [Accessed 1 Dec. 2019].

[2]"JupyterLab Documentation — JupyterLab 1.2.6 documentation", *Jupyterlab.readthedocs.io*, 2020. [Online]. Available: https://jupyterlab.readthedocs.io/en/stable/. [Accessed: 02- Dec- 2019].

[3]"scikit-learn: machine learning in Python — scikit-learn 0.22.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 01- Dec- 2019].

[4]S. Srinidhi, "How to split your dataset to train and test datasets using SciKit Learn", *Medium*, 2020. [Online]. Available: https://medium.com/@contactsunny/how-to-split-your-dataset-to-train-and-test-datasets-using-scikit-learn-e7cf6eb5e0d. [Accessed: 01- Dec- 2020].

[5]A. Navlani, "Naive Bayes Classification using Scikit-learn", *DataCamp Community*, 2020. [Online]. Available: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn. [Accessed: 02- Dec-2019].

[6]S. Li, "Building A Logistic Regression in Python, Step by Step", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8. [Accessed: 01- Dec- 2019].

[7]W. Koehrsen, "An Implementation and Explanation of the Random Forest in Python", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76. [Accessed: 01- Dec- 2019].