Features to Extract/Transform:

Idx	Feature Name	Description	Extraction/Transformation Method
1	JPG_P	% of HD – JPGs	Fiwalk – file extensions for allocated files; Sceadan type 32 for unallocated blocks [a]
2	Vid_P	% of HD – video files	Fiwalk – file extensions for allocated files; Sceadan types 35-39, 42 for unallocated blocks *
3	JPG-vid_P	Sum of two items above	N/A – simple addition of other extracted/transformed data
4	Rand_P	% of HD tagged 'random'	Scan_bulk[b]
5	Alloc_P	% of partitions allocated	Fiwalk[c]
100	Email_totfreq	Total *non-unique* instances of email addresses found	Scan_email; Sum of all N= in email_histogram (or "email" count in report.xml)
101	Email_totuniq	Total *unique* email addresses found	Scan_email; Number of "N=" rows in email_histogram
102	Edom_totuniq	Total number of *unique* email *domains* found	Scan_email; Parse email_histogram for unique domains, report number of unique domains
103	Edom01_totfreq	Total instances in 1st most frequently occurring email address domain	Scan_email; Parse email_histogram for unique domains, add N= values for unique domains, value of 1st highest value
104	Edom02_ totfreq	Total instances in 2nd most frequently occurring email address domain	Scan_email; Parse email_histogram for unique domains, add N= values for unique domains, value of 2nd highest value
105	Edom03_ totfreq	.	.
106	Edom04_ totfreq	.	.
107	Edom05_ totfreq	.	.
108	Edom06_ totfreq	.	.
109	Edom07_ totfreq	.	.
110	Edom08_ totfreq	.	.
111	Edom09_ totfreq	.	.
112	Edom10_ totfreq	.	.
113	Edom-outorg_P	Total instances of email addresses	Scan_email; Parse email_histogram for unique domains, add N= values

		for domains *outside* primary user's org	for domains other than user specified domain of disk's org
200	CCN_totfreq	Total *non-unique* instances of CCNs found	Scan_accts; Sum of all N= in ccn_histogram where num_length>11
201	CCN_totuniq	Total *unique* CCNs found	Scan_accts; Number of "N=" rows in ccn_histogram where num_length> 11
202	CCN01_totfreq	Total instances in 1st most frequently occurring CCN	Scan_accts; ccn_histogram file; 1st highest N= value, where num_length>11
203	CCN02_ totfreq	Total instances in 2nd most frequently occurring CCN	Scan_accts; ccn_histogram file; 2nd highest N= value…
204	CCN03_ totfreq	.	.
205	CCN04_ totfreq	.	.
206	CCN05_ totfreq	.	.
207	CCN06_ totfreq	.	.
208	CCN07_ totfreq	.	.
209	CCN08_ totfreq	.	.
210	CCN09_ totfreq	.	.
211	CCN10_ totfreq	.	.
300	SSN_totfreq	Total *non-unique* instances of SSNs found	Scan_accts; Sum of all N= in ccn_histogram where num_length is 9-11
301	SSN_totuniq	Total *unique* SSNs found	Scan_accts; Number of "N=" rows in ccn_histogram where num_length is 9-11
302	SSN01_totfreq	Total instances in 1st most frequently occurring SSN	Scan_accts; ccn_histogram file; 1st highest N= value, where num_length is 9-11
303	SSN02_ totfreq	Total instances in 2nd most frequently occurring SSN	Scan_accts; ccn_histogram file; 2nd highest N= value…
304	SSN03_ totfreq	.	.
305	SSN04_ totfreq	.	.
306	SSN05_ totfreq	.	.
307	SSN06_ totfreq	.	.
308	SSN07_ totfreq	.	.
309	SSN08_ totfreq	.	.
310	SSN09_ totfreq	.	.

311	SSN10_ totfreq	.	.
400	URL_totfreq	Total *non-unique* URLs	Sum of all N= in url_histogram (or "URL" count in report.xml)
401	URL_totuniq	Total *unique* URLs found	Number of "N=" rows in url_histogram
402	Udom_totuniq	Total number of *unique* URLs at domain level found	Parse url_histogram for unique domains, report number of unique domains
403	Udom01_totfreq	Total instances in 1st most frequently occurring URL at domain level	Parse url_histogram for unique domains, add N= values for unique domains, value of 1st highest value
404	Udom 02_ totfreq	Total instances in 2nd most frequently occurring URL at domain level	Parse url_histogram for unique domains, add N= values for unique domains, value of 1st highest value
405	Udom 03_ totfreq		.
406	Udom 04_ totfreq		.
407	Udom 05_ totfreq		.
408	Udom 06_ totfreq		.
409	Udom 07_ totfreq		.
410	Udom 08_ totfreq		.
411	Udom 09_ totfreq		.
412	Udom 10_ totfreq		.
500	Other_P	File type not covered in DB	TSK/fiwalk file system walk – file extensions for allocated files; Sceadan for unallocated blocks
501	Text_P	Sceadan types 1, 3, .txt, .log	.
505	ASP_P	Sceadan type 6, .asp, .aspx	.
509	CSS_P	Sceadan type 10, .css	.
510	B64_P	Sceadan type 11	.
511	B85_P	Sceadan type 12	.
512	B16_P	Sceadan type 13	.
513	URLencoded_P	Sceadan type 14	.
514	PS_P	Sceadan type 15	.
516	Email_P	Sceadan type 17, 18, .pst, .ost, .pab, .msf	.
517	PNG_P	Sceadan type 19, .png	.

518	TIF_P	Sceadan type 21, .tif, .tiff	.
519	JB2_P	Sceadan type 22, .jb2, .jbig, .jbig2	.
520	Zip_P	Sceadan type 23, 24, 27, .gz, .gzip, .tgz, .z, .taz, .zip, .bz2, bzip, bzip2	.
522	RPM_P	Sceadan type 26, .rpm	.
523	PDF_P	Sceadan type 28, .pdf	.
527	Audio_P	Sceadan types 33, 34, 40, 41 .mp3, .m4a, .aac, .wav, .wma	.
531	EXE_P	Sceadan type 49, .exe	.
532	DLL_P	Sceadan type 50, .dll	.
533	ELF_P	Sceadan type 51, .elf	.
534	BMP_P	Sceadan type 52, .bmp	.
535	GIF_P	Sceadan type 20, .gif	.
536	WinSys_P	.inf .pnf .mof .sys .msi .cfg .chm .cab .com .hlp .msc .sdb .fon .cur .ax .ttf .query .ver .ott .cat .xcu .nls .state, .dlg, .font	.
537	Binary_P	.bin, .dat	.
538	Dev_P	Sceadan type 8, 9, 25, .js, .py, .pl, .c, .cpp, .h, .lib, .tcl, .idx, .java, .jar, .class, .pm, .sh	.
539	Ini_P	.ini	.
540	Lnk_P	.lnk	.
541	Tmp_P	.tmp	.
542	Spreadsheet_P	Sceadan type 2, 30, 44, .csv, .xlsx, .xls, .ods	.
543	Markup_P	Sceadan type 4, 5, 7, .html, .htm, .xml, .json, .dtd	.
544	WordProc_P	Sceadan type 16, 29, 43, .rtf, .docx, .doc, .odt	.
545	Present_P	Sceadan type 31, 45, .pptx, .ppt, .odp	.

NOTE: The addition of types 542-545 consolidate (and therefore remove) features 502-504, 506-508, 515, 521, 524-526, 528-530.

[a] Calculate percentage by dividing sum of (all allocated .JPG file sizes, plus sum of all unallocated blocks classified as JPG (count * block size)) by partition size (Fiwalk (block_size * block_count))

[b] Calculate percentage by dividing the number of blocks tagged ‘random’ by the number of blocks scanned (# of sbuf; disk size/sbuf_size)

[c] Simple percentage reported by Fiwalk output. Sum ‘filesize’ of all ‘Alloc: 1’ and ‘name_type: r’entries and divide by partition size (block_size * block_count); account for multiple file system partitions where appropriate. Ignore any with ‘Unalloc: 1’ and/or ‘name_type: d’.