

ALESSI PI

**ASTRO PI PROJECT – MISSION SPACE LAB
LICEO SCIENTIFICO “G. ALESSI”
PERUGIA, ITALY**



ASTRO PI





CHI SIAMO?





IL NOSTRO PROGETTO

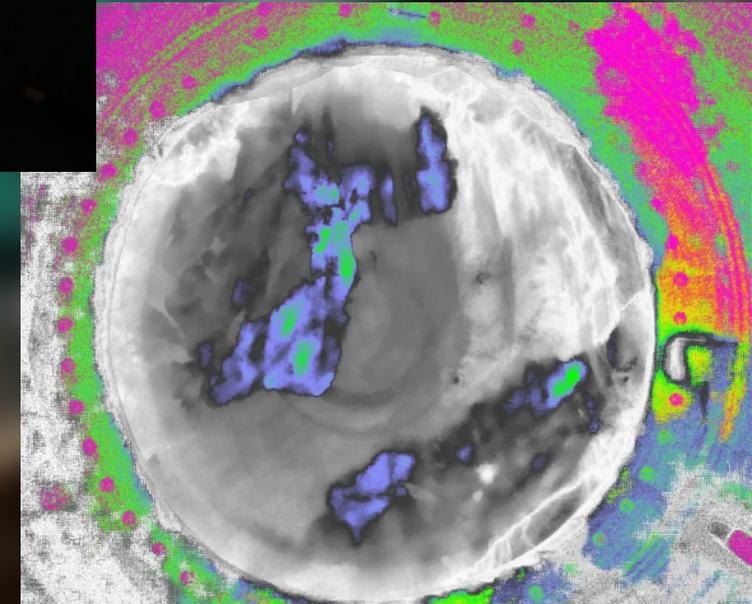
L'esperienza prevedeva di indagare sulla salute delle piante, misurando il Normalized Difference Vegetation Index (NDVI). Ci aspettavamo una variazione dell'NDVI sulla superficie terrestre in relazione alla latitudine e al valore del campo magnetico.

```

1 import sys, os, math, time, datetime, numpy as np
2 from sklearn import svm
3 from sklearn import svm
4 from sklearn import svm
5 from sklearn import svm
6 class ColorMap(object):
7     def __init__(self, cmap):
8         self.cmap = cmap
9         self.colors = self.cmap.colors
10        self.norm = self.cmap.norm
11        self.vmin = self.cmap.vmin
12        self.vmax = self.cmap.vmax
13        self.reverse = self.cmap.reverse
14
15    def __call__(self, v):
16        if v < self.vmin:
17            return self.colors[0]
18        if v > self.vmax:
19            return self.colors[-1]
20        return self.colors[int((v - self.vmin) / (self.vmax - self.vmin) * (len(self.colors) - 1))]
21
22    def __getitem__(self, i):
23        return self.colors[i]
24
25    def __iter__(self):
26        return iter(self.colors)
27
28    def __len__(self):
29        return len(self.colors)
30
31    def __repr__(self):
32        return "ColorMap(%s)" % self.cmap
33
34    def __str__(self):
35        return "ColorMap(%s)" % self.cmap
36
37    def __eq__(self, other):
38        return self.cmap == other.cmap
39
40    def __neq__(self, other):
41        return self.cmap != other.cmap
42
43    def __hash__(self):
44        return hash(self.cmap)
45
46    def __copy__(self):
47        return ColorMap(self.cmap)
48
49    def __deepcopy__(self, memo):
50        return ColorMap(self.cmap)
51
52    def __del__(self):
53        pass
54
55    def __getstate__(self):
56        return self.cmap
57
58    def __setstate__(self, state):
59        self.cmap = state
60
61    def __getattr__(self, attr):
62        return getattr(self.cmap, attr)
63
64    def __setattr__(self, attr, value):
65        setattr(self.cmap, attr, value)
66
67    def __delattr__(self, attr):
68        delattr(self.cmap, attr)
69
70    def __dir__(self):
71        return dir(self.cmap)
72
73    def __contains__(self, item):
74        return item in self.cmap
75
76    def __getitem__(self, item):
77        return self.cmap[item]
78
79    def __setitem__(self, item, value):
80        self.cmap[item] = value
81
82    def __delitem__(self, item):
83        del self.cmap[item]
84
85    def __iter__(self):
86        return iter(self.cmap)
87
88    def __len__(self):
89        return len(self.cmap)
90
91    def __repr__(self):
92        return "ColorMap(%s)" % self.cmap
93
94    def __str__(self):
95        return "ColorMap(%s)" % self.cmap
96
97    def __eq__(self, other):
98        return self.cmap == other.cmap
99
100   def __neq__(self, other):
101       return self.cmap != other.cmap
102
103   def __hash__(self):
104       return hash(self.cmap)
105
106   def __copy__(self):
107       return ColorMap(self.cmap)
108
109   def __deepcopy__(self, memo):
110       return ColorMap(self.cmap)
111
112   def __del__(self):
113       pass
114
115   def __getstate__(self):
116       return self.cmap
117
118   def __setstate__(self, state):
119       self.cmap = state
120
121   def __getattr__(self, attr):
122       return getattr(self.cmap, attr)
123
124   def __setattr__(self, attr, value):
125       setattr(self.cmap, attr, value)
126
127   def __delattr__(self, attr):
128       delattr(self.cmap, attr)
129
130   def __dir__(self):
131       return dir(self.cmap)
132
133   def __contains__(self, item):
134       return item in self.cmap
135
136   def __getitem__(self, item):
137       return self.cmap[item]
138
139   def __setitem__(self, item, value):
140       self.cmap[item] = value
141
142   def __delitem__(self, item):
143       del self.cmap[item]
144
145   def __iter__(self):
146       return iter(self.cmap)
147
148   def __len__(self):
149       return len(self.cmap)
150
151   def __repr__(self):
152       return "ColorMap(%s)" % self.cmap
153
154   def __str__(self):
155       return "ColorMap(%s)" % self.cmap
156
157   def __eq__(self, other):
158       return self.cmap == other.cmap
159
160   def __neq__(self, other):
161       return self.cmap != other.cmap
162
163   def __hash__(self):
164       return hash(self.cmap)
165
166   def __copy__(self):
167       return ColorMap(self.cmap)
168
169   def __deepcopy__(self, memo):
170       return ColorMap(self.cmap)
171
172   def __del__(self):
173       pass
174
175   def __getstate__(self):
176       return self.cmap
177
178   def __setstate__(self, state):
179       self.cmap = state
180
181   def __getattr__(self, attr):
182       return getattr(self.cmap, attr)
183
184   def __setattr__(self, attr, value):
185       setattr(self.cmap, attr, value)
186
187   def __delattr__(self, attr):
188       delattr(self.cmap, attr)
189
190   def __dir__(self):
191       return dir(self.cmap)
192
193   def __contains__(self, item):
194       return item in self.cmap
195
196   def __getitem__(self, item):
197       return self.cmap[item]
198
199   def __setitem__(self, item, value):
200       self.cmap[item] = value
201
202   def __delitem__(self, item):
203       del self.cmap[item]
204
205   def __iter__(self):
206       return iter(self.cmap)
207
208   def __len__(self):
209       return len(self.cmap)
210
211   def __repr__(self):
212       return "ColorMap(%s)" % self.cmap
213
214   def __str__(self):
215       return "ColorMap(%s)" % self.cmap
216
217   def __eq__(self, other):
218       return self.cmap == other.cmap
219
220   def __neq__(self, other):
221       return self.cmap != other.cmap
222
223   def __hash__(self):
224       return hash(self.cmap)
225
226   def __copy__(self):
227       return ColorMap(self.cmap)
228
229   def __deepcopy__(self, memo):
230       return ColorMap(self.cmap)
231
232   def __del__(self):
233       pass
234
235   def __getstate__(self):
236       return self.cmap
237
238   def __setstate__(self, state):
239       self.cmap = state
240
241   def __getattr__(self, attr):
242       return getattr(self.cmap, attr)
243
244   def __setattr__(self, attr, value):
245       setattr(self.cmap, attr, value)
246
247   def __delattr__(self, attr):
248       delattr(self.cmap, attr)
249
250   def __dir__(self):
251       return dir(self.cmap)
252
253   def __contains__(self, item):
254       return item in self.cmap
255
256   def __getitem__(self, item):
257       return self.cmap[item]
258
259   def __setitem__(self, item, value):
260       self.cmap[item] = value
261
262   def __delitem__(self, item):
263       del self.cmap[item]
264
265   def __iter__(self):
266       return iter(self.cmap)
267
268   def __len__(self):
269       return len(self.cmap)
270
271   def __repr__(self):
272       return "ColorMap(%s)" % self.cmap
273
274   def __str__(self):
275       return "ColorMap(%s)" % self.cmap
276
277   def __eq__(self, other):
278       return self.cmap == other.cmap
279
280   def __neq__(self, other):
281       return self.cmap != other.cmap
282
283   def __hash__(self):
284       return hash(self.cmap)
285
286   def __copy__(self):
287       return ColorMap(self.cmap)
288
289   def __deepcopy__(self, memo):
290       return ColorMap(self.cmap)
291
292   def __del__(self):
293       pass
294
295   def __getstate__(self):
296       return self.cmap
297
298   def __setstate__(self, state):
299       self.cmap = state
300
301   def __getattr__(self, attr):
302       return getattr(self.cmap, attr)
303
304   def __setattr__(self, attr, value):
305       setattr(self.cmap, attr, value)
306
307   def __delattr__(self, attr):
308       delattr(self.cmap, attr)
309
310   def __dir__(self):
311       return dir(self.cmap)
312
313   def __contains__(self, item):
314       return item in self.cmap
315
316   def __getitem__(self, item):
317       return self.cmap[item]
318
319   def __setitem__(self, item, value):
320       self.cmap[item] = value
321
322   def __delitem__(self, item):
323       del self.cmap[item]
324
325   def __iter__(self):
326       return iter(self.cmap)
327
328   def __len__(self):
329       return len(self.cmap)
330
331   def __repr__(self):
332       return "ColorMap(%s)" % self.cmap
333
334   def __str__(self):
335       return "ColorMap(%s)" % self.cmap
336
337   def __eq__(self, other):
338       return self.cmap == other.cmap
339
340   def __neq__(self, other):
341       return self.cmap != other.cmap
342
343   def __hash__(self):
344       return hash(self.cmap)
345
346   def __copy__(self):
347       return ColorMap(self.cmap)
348
349   def __deepcopy__(self, memo):
350       return ColorMap(self.cmap)
351
352   def __del__(self):
353       pass
354
355   def __getstate__(self):
356       return self.cmap
357
358   def __setstate__(self, state):
359       self.cmap = state
360
361   def __getattr__(self, attr):
362       return getattr(self.cmap, attr)
363
364   def __setattr__(self, attr, value):
365       setattr(self.cmap, attr, value)
366
367   def __delattr__(self, attr):
368       delattr(self.cmap, attr)
369
370   def __dir__(self):
371       return dir(self.cmap)
372
373   def __contains__(self, item):
374       return item in self.cmap
375
376   def __getitem__(self, item):
377       return self.cmap[item]
378
379   def __setitem__(self, item, value):
380       self.cmap[item] = value
381
382   def __delitem__(self, item):
383       del self.cmap[item]
384
385   def __iter__(self):
386       return iter(self.cmap)
387
388   def __len__(self):
389       return len(self.cmap)
390
391   def __repr__(self):
392       return "ColorMap(%s)" % self.cmap
393
394   def __str__(self):
395       return "ColorMap(%s)" % self.cmap
396
397   def __eq__(self, other):
398       return self.cmap == other.cmap
399
400   def __neq__(self, other):
401       return self.cmap != other.cmap
402
403   def __hash__(self):
404       return hash(self.cmap)
405
406   def __copy__(self):
407       return ColorMap(self.cmap)
408
409   def __deepcopy__(self, memo):
410       return ColorMap(self.cmap)
411
412   def __del__(self):
413       pass
414
415   def __getstate__(self):
416       return self.cmap
417
418   def __setstate__(self, state):
419       self.cmap = state
420
421   def __getattr__(self, attr):
422       return getattr(self.cmap, attr)
423
424   def __setattr__(self, attr, value):
425       setattr(self.cmap, attr, value)
426
427   def __delattr__(self, attr):
428       delattr(self.cmap, attr)
429
430   def __dir__(self):
431       return dir(self.cmap)
432
433   def __contains__(self, item):
434       return item in self.cmap
435
436   def __getitem__(self, item):
437       return self.cmap[item]
438
439   def __setitem__(self, item, value):
440       self.cmap[item] = value
441
442   def __delitem__(self, item):
443       del self.cmap[item]
444
445   def __iter__(self):
446       return iter(self.cmap)
447
448   def __len__(self):
449       return len(self.cmap)
450
451   def __repr__(self):
452       return "ColorMap(%s)" % self.cmap
453
454   def __str__(self):
455       return "ColorMap(%s)" % self.cmap
456
457   def __eq__(self, other):
458       return self.cmap == other.cmap
459
460   def __neq__(self, other):
461       return self.cmap != other.cmap
462
463   def __hash__(self):
464       return hash(self.cmap)
465
466   def __copy__(self):
467       return ColorMap(self.cmap)
468
469   def __deepcopy__(self, memo):
470       return ColorMap(self.cmap)
471
472   def __del__(self):
473       pass
474
475   def __getstate__(self):
476       return self.cmap
477
478   def __setstate__(self, state):
479       self.cmap = state
480
481   def __getattr__(self, attr):
482       return getattr(self.cmap, attr)
483
484   def __setattr__(self, attr, value):
485       setattr(self.cmap, attr, value)
486
487   def __delattr__(self, attr):
488       delattr(self.cmap, attr)
489
490   def __dir__(self):
491       return dir(self.cmap)
492
493   def __contains__(self, item):
494       return item in self.cmap
495
496   def __getitem__(self, item):
497       return self.cmap[item]
498
499   def __setitem__(self, item, value):
500       self.cmap[item] = value
501
502   def __delitem__(self, item):
503       del self.cmap[item]
504
505   def __iter__(self):
506       return iter(self.cmap)
507
508   def __len__(self):
509       return len(self.cmap)
510
511   def __repr__(self):
512       return "ColorMap(%s)" % self.cmap
513
514   def __str__(self):
515       return "ColorMap(%s)" % self.cmap
516
517   def __eq__(self, other):
518       return self.cmap == other.cmap
519
520   def __neq__(self, other):
521       return self.cmap != other.cmap
522
523   def __hash__(self):
524       return hash(self.cmap)
525
526   def __copy__(self):
527       return ColorMap(self.cmap)
528
529   def __deepcopy__(self, memo):
530       return ColorMap(self.cmap)
531
532   def __del__(self):
533       pass
534
535   def __getstate__(self):
536       return self.cmap
537
538   def __setstate__(self, state):
539       self.cmap = state
540
541   def __getattr__(self, attr):
542       return getattr(self.cmap, attr)
543
544   def __setattr__(self, attr, value):
545       setattr(self.cmap, attr, value)
546
547   def __delattr__(self, attr):
548       delattr(self.cmap, attr)
549
550   def __dir__(self):
551       return dir(self.cmap)
552
553   def __contains__(self, item):
554       return item in self.cmap
555
556   def __getitem__(self, item):
557       return self.cmap[item]
558
559   def __setitem__(self, item, value):
560       self.cmap[item] = value
561
562   def __delitem__(self, item):
563       del self.cmap[item]
564
565   def __iter__(self):
566       return iter(self.cmap)
567
568   def __len__(self):
569       return len(self.cmap)
570
571   def __repr__(self):
572       return "ColorMap(%s)" % self.cmap
573
574   def __str__(self):
575       return "ColorMap(%s)" % self.cmap
576
577   def __eq__(self, other):
578       return self.cmap == other.cmap
579
580   def __neq__(self, other):
581       return self.cmap != other.cmap
582
583   def __hash__(self):
584       return hash(self.cmap)
585
586   def __copy__(self):
587       return ColorMap(self.cmap)
588
589   def __deepcopy__(self, memo):
590       return ColorMap(self.cmap)
591
592   def __del__(self):
593       pass
594
595   def __getstate__(self):
596       return self.cmap
597
598   def __setstate__(self, state):
599       self.cmap = state
600
601   def __getattr__(self, attr):
602       return getattr(self.cmap, attr)
603
604   def __setattr__(self, attr, value):
605       setattr(self.cmap, attr, value)
606
607   def __delattr__(self, attr):
608       delattr(self.cmap, attr)
609
610   def __dir__(self):
611       return dir(self.cmap)
612
613   def __contains__(self, item):
614       return item in self.cmap
615
616   def __getitem__(self, item):
617       return self.cmap[item]
618
619   def __setitem__(self, item, value):
620       self.cmap[item] = value
621
622   def __delitem__(self, item):
623       del self.cmap[item]
624
625   def __iter__(self):
626       return iter(self.cmap)
627
628   def __len__(self):
629       return len(self.cmap)
630
631   def __repr__(self):
632       return "ColorMap(%s)" % self.cmap
633
634   def __str__(self):
635       return "ColorMap(%s)" % self.cmap
636
637   def __eq__(self, other):
638       return self.cmap == other.cmap
639
640   def __neq__(self, other):
641       return self.cmap != other.cmap
642
643   def __hash__(self):
644       return hash(self.cmap)
645
646   def __copy__(self):
647       return ColorMap(self.cmap)
648
649   def __deepcopy__(self, memo):
650       return ColorMap(self.cmap)
651
652   def __del__(self):
653       pass
654
655   def __getstate__(self):
656       return self.cmap
657
658   def __setstate__(self, state):
659       self.cmap = state
660
661   def __getattr__(self, attr):
662       return getattr(self.cmap, attr)
663
664   def __setattr__(self, attr, value):
665       setattr(self.cmap, attr, value)
666
667   def __delattr__(self, attr):
668       delattr(self.cmap, attr)
669
670   def __dir__(self):
671       return dir(self.cmap)
672
673   def __contains__(self, item):
674       return item in self.cmap
675
676   def __getitem__(self, item):
677       return self.cmap[item]
678
679   def __setitem__(self, item, value):
680       self.cmap[item] = value
681
682   def __delitem__(self, item):
683       del self.cmap[item]
684
685   def __iter__(self):
686       return iter(self.cmap)
687
688   def __len__(self):
689       return len(self.cmap)
690
691   def __repr__(self):
692       return "ColorMap(%s)" % self.cmap
693
694   def __str__(self):
695       return "ColorMap(%s)" % self.cmap
696
697   def __eq__(self, other):
698       return self.cmap == other.cmap
699
700   def __neq__(self, other):
701       return self.cmap != other.cmap
702
703   def __hash__(self):
704       return hash(self.cmap)
705
706   def __copy__(self):
707       return ColorMap(self.cmap)
708
709   def __deepcopy__(self, memo):
710       return ColorMap(self.cmap)
711
712   def __del__(self):
713       pass
714
715   def __getstate__(self):
716       return self.cmap
717
718   def __setstate__(self, state):
719       self.cmap = state
720
721   def __getattr__(self, attr):
722       return getattr(self.cmap, attr)
723
724   def __setattr__(self, attr, value):
725       setattr(self.cmap, attr, value)
726
727   def __delattr__(self, attr):
728       delattr(self.cmap, attr)
729
730   def __dir__(self):
731       return dir(self.cmap)
732
733   def __contains__(self, item):
734       return item in self.cmap
735
736   def __getitem__(self, item):
737       return self.cmap[item]
738
739   def __setitem__(self, item, value):
740       self.cmap[item] = value
741
742   def __delitem__(self, item):
743       del self.cmap[item]
744
745   def __iter__(self):
746       return iter(self.cmap)
747
748   def __len__(self):
749       return len(self.cmap)
750
751   def __repr__(self):
752       return "ColorMap(%s)" % self.cmap
753
754   def __str__(self):
755       return "ColorMap(%s)" % self.cmap
756
757   def __eq__(self, other):
758       return self.cmap == other.cmap
759
760   def __neq__(self, other):
761       return self.cmap != other.cmap
762
763   def __hash__(self):
764       return hash(self.cmap)
765
766   def __copy__(self):
767       return ColorMap(self.cmap)
768
769   def __deepcopy__(self, memo):
770       return ColorMap(self.cmap)
771
772   def __del__(self):
773       pass
774
775   def __getstate__(self):
776       return self.cmap
777
778   def __setstate__(self, state):
779       self.cmap = state
780
781   def __getattr__(self, attr):
782       return getattr(self.cmap, attr)
783
784   def __setattr__(self, attr, value):
785       setattr(self.cmap, attr, value)
786
787   def __delattr__(self, attr):
788       delattr(self.cmap, attr)
789
790   def __dir__(self):
791       return dir(self.cmap)
792
793   def __contains__(self, item):
794       return item in self.cmap
795
796   def __getitem__(self, item):
797       return self.cmap[item]
798
799   def __setitem__(self, item, value):
800       self.cmap[item] = value
801
802   def __delitem__(self, item):
803       del self.cmap[item]
804
805   def __iter__(self):
806       return iter(self.cmap)
807
808   def __len__(self):
809       return len(self.cmap)
810
811   def __repr__(self):
812       return "ColorMap(%s)" % self.cmap
813
814   def __str__(self):
815       return "ColorMap(%s)" % self.cmap
816
817   def __eq__(self, other):
818       return self.cmap == other.cmap
819
820   def __neq__(self, other):
821       return self.cmap != other.cmap
822
823   def __hash__(self):
824       return hash(self.cmap)
825
826   def __copy__(self):
827       return ColorMap(self.cmap)
828
829   def __deepcopy__(self, memo):
830       return ColorMap(self.cmap)
831
832   def __del__(self):
833       pass
834
835   def __getstate__(self):
836       return self.cmap
837
838   def __setstate__(self, state):
839       self.cmap = state
840
841   def __getattr__(self, attr):
842       return getattr(self.cmap, attr)
843
844   def __setattr__(self, attr, value):
845       setattr(self.cmap, attr, value)
846
847   def __delattr__(self, attr):
848       delattr(self.cmap, attr)
849
850   def __dir__(self):
851       return dir(self.cmap)
852
853   def __contains__(self, item):
854       return item in self.cmap
855
856   def __getitem__(self, item):
857       return self.cmap[item]
858
859   def __setitem__(self, item, value):
860       self.cmap[item] = value
861
862   def __delitem__(self, item):
863       del self.cmap[item]
864
865   def __iter__(self):
866       return iter(self.cmap)
867
868   def __len__(self):
869       return len(self.cmap)
870
871   def __repr__(self):
872       return "ColorMap(%s)" % self.cmap
873
874   def __str__(self):
875       return "ColorMap(%s)" % self.cmap
876
877   def __eq__(self, other):
878       return self.cmap == other.cmap
879
880   def __neq__(self, other):
881       return self.cmap != other.cmap
882
883   def __hash__(self):
884       return hash(self.cmap)
885
886   def __copy__(self):
887       return ColorMap(self.cmap)
888
889   def __deepcopy__(self, memo):
890       return ColorMap(self.cmap)
891
892   def __del__(self):
893       pass
894
895   def __getstate__(self):
896       return self.cmap
897
898   def __setstate__(self, state):
899       self.cmap = state
900
901   def __getattr__(self, attr):
902       return getattr(self.cmap, attr)
903
904   def __setattr__(self, attr, value):
905       setattr(self.cmap, attr, value)
906
907   def __delattr__(self, attr):
908       delattr(self.cmap, attr)
909
910   def __dir__(self):
911       return dir(self.cmap)
912
913   def __contains__(self, item):
914       return item in self.cmap
915
916   def __getitem__(self, item):
917       return self.cmap[item]
918
919   def __setitem__(self, item, value):
920       self.cmap[item] = value
921
922   def __delitem__(self, item):
923       del self.cmap[item]
924
925   def __iter__(self):
926       return iter(self.cmap)
927
928   def __len__(self):
929       return len(self.cmap)
930
931   def __repr__(self):
932       return "ColorMap(%s)" % self.cmap
933
934   def __str__(self):
935       return "ColorMap(%s)" % self.cmap
936
937   def __eq__(self, other):
938       return self.cmap == other.cmap
939
940   def __neq__(self, other):
941       return self.cmap != other.cmap
942
943   def __hash__(self):
944       return hash(self.cmap)
945
946   def __copy__(self):
947       return ColorMap(self.cmap)
948
949   def __deepcopy__(self, memo):
950       return ColorMap(self.cmap)
951
952   def __del__(self):
953       pass
954
955   def __getstate__(self):
956       return self.cmap
957
958   def __setstate__(self, state):
959       self.cmap = state
960
961   def __getattr__(self, attr):
962       return getattr(self.cmap, attr)
963
964   def __setattr__(self, attr, value):
965       setattr(self.cmap, attr, value)
966
967   def __delattr__(self, attr):
968       delattr(self.cmap, attr)
969
970   def __dir__(self):
971       return dir(self.cmap)
972
973   def __contains__(self, item):
974       return item in self.cmap
975
976   def __getitem__(self, item):
977       return self.cmap[item]
978
979   def __setitem__(self, item, value):
980       self.cmap[item] = value
981
982   def __delitem__(self, item):
983       del self.cmap[item]
984
985   def __iter__(self):
986       return iter(self.cmap)
987
988   def __len__(self):
989       return len(self.cmap)
990
991   def __repr__(self):
992       return "ColorMap(%s)" % self.cmap
993
994   def __str__(self):
995       return "ColorMap(%s)" % self.cmap
996
997   def __eq__(self, other):
998       return self.cmap == other.cmap
999
1000  def __neq__(self, other):
1001      return self.cmap != other.cmap
1002
1003  def __hash__(self):
1004      return hash(self.cmap)
1005
1006  def __copy__(self):
1007      return ColorMap(self.cmap)
1008
1009  def __deepcopy__(self, memo):
1010      return ColorMap(self.cmap)
1011
1012  def __del__(self):
1013      pass
1014
1015  def __getstate__(self):
1016      return self.cmap
1017
1018  def __setstate__(self, state):
1019      self.cmap = state
1020
1021  def __getattr__(self, attr):
1022      return getattr(self.cmap, attr)
1023
1024  def __setattr__(self, attr, value):
1025      setattr(self.cmap, attr, value)
1026
1027  def __delattr__(self, attr):
1028      delattr(self.cmap, attr)
1029
1030  def __dir__(self):
1031      return dir(self.cmap)
1032
1033  def __contains__(self, item):
1034      return item in self.cmap
1035
1036  def __getitem__(self, item):
1037      return self.cmap[item]
1038
1039  def __setitem__(self, item, value):
1040      self.cmap[item] = value
1041
1042  def __delitem__(self, item):
1043      del self.cmap[item]
1044
1045  def __iter__(self):
1046      return iter(self.cmap)
1047
1048  def __len__(self):
1049      return len(self.cmap)
1050
1051  def __repr__(self):
1052      return "ColorMap(%s)" % self.cmap
1053
1054  def __str__(self):
1055      return "ColorMap(%s)" % self.cmap
1056
1057  def __eq__(self, other):
1058      return self.cmap == other.cmap
1059
1060  def __neq__(self, other):
1061      return self.cmap != other.cmap
1062
1063  def __hash__(self):
1064      return hash(self.cmap)
1065
1066  def __copy__(self):
1067      return ColorMap(self.cmap)
1068
1069  def __deepcopy__(self, memo):
1070      return ColorMap(self.cmap)
1071
1072  def __del__(self):
1073      pass
1074
1075  def __getstate__(self):
1076      return self.cmap
1077
1078  def __setstate__(self, state):
1079      self.cmap = state
1080
1081  def __getattr__(self, attr):
1082      return getattr(self.cmap, attr)
1083
1084  def __setattr__(self, attr, value):
1085      setattr(self.cmap, attr, value)
1086
1087  def __delattr__(self, attr):
1088      delattr(self.cmap, attr)
1089
1090  def __dir__(self):
1091      return dir(self.cmap)
1092
1093  def __contains__(self, item):
1094      return item in self.cmap
1095
1096  def __getitem__(self, item):
1097      return self.cmap[item]
1098
1099  def __setitem__(self, item, value):
1100      self.cmap[item] = value
1101
1102  def __delitem__(self, item):
1103      del self.cmap[item]
1104
1105  def __iter__(self):
1106      return iter(self.cmap)
1107
1108  def __len__(self):
1109      return len(self.cmap)
1110
1111  def __repr__(self):
1112      return "ColorMap(%s)" % self.cmap
1113
1114  def __str__(self):
1115      return "ColorMap(%s)" % self.cmap
1116
1117  def __eq__(self, other):
1118      return self.cmap == other.cmap
1119
1120  def __neq__(self, other):
1121      return self.cmap != other.cmap
1122
1123  def __hash__(self):
1124      return hash(self.cmap)
1125
1126  def __copy__(self):
1127      return ColorMap(self.cmap)
1128
1129  def __deepcopy__(self, memo):
1130      return ColorMap(self.cmap)
1131
1132  def __del__(self):
1133      pass
1134
1135  def __getstate__(self):
1136      return self.cmap
1137
1138  def __setstate__(self, state):
1139      self.cmap = state
1140
1141  def __getattr__(self, attr):
1142      return getattr(self.cmap, attr)
1143
1144  def __setattr__(self, attr, value):
1145      setattr(self.cmap, attr, value)
1146
1147  def __delattr__(self, attr):
1148      delattr(self.cmap, attr)
1149
1150  def __dir__(self):
1151      return dir(self.cmap)
1152
1153  def __contains__(self, item):
1154      return item in self.cmap
1155
1156  def __getitem__(self, item):
1157      return self.cmap[item]
1158
1159  def __setitem__(self, item, value):
1160      self.cmap[item] = value
1161
1162  def __delitem__(self, item):
1163      del self.cmap[item]
1164
1165  def __iter__(self):
1166      return iter(self.cmap)
1167
1168  def __len__(self):
1169      return len(self.cmap)
1170
1171  def __repr__(self):
1172      return "ColorMap(%s)" % self.cmap
1173
1174  def __str__(self):
1175      return "ColorMap(%s)" % self.cmap
1176
1177  def __eq__(self, other):
1178      return self.cmap == other.cmap
1179
1180  def __neq__(self, other):
1181      return self.cmap != other.cmap
1182
1183  def __hash__(self):
1184      return hash(self.cmap)
1185
1186  def __copy__(self):
1187      return ColorMap(self.cmap)
1188
1189  def __deepcopy__(self, memo):
1190      return ColorMap(self.cmap)
1191
1192  def __del__(self):
1193      pass
1194
1195  def __getstate__(self):
1196      return self.cmap
1197
1198  def __setstate__(self, state):
1199      self.cmap = state
1200
1201  def __getattr__(self, attr):
1202      return getattr(self.cmap, attr)
1203
1204  def __setattr__(self, attr, value):
1205      setattr(self.cmap, attr, value)
1206
1207  def __delattr__(self, attr):
1208      delattr(self.cmap, attr)
1209
1210  def __dir__(self):
1211      return dir(self.cmap)
1212
1213  def __contains__(self, item):
1214      return item in self.cmap
1215
1216  def __getitem__(self, item):
1217      return self.cmap[item]
1218
1219  def __setitem__(self, item, value):
1220      self.cmap[item] = value
1221
1222  def __delitem__(self, item):
1223      del self.cmap[item]
1224
1225  def __iter__(self):
1226      return iter(self.cmap)
1227
1228  def __len__(self):
1229      return len(self.cmap)
1230
1231  def __repr__(self):
1232      return "ColorMap(%s)" % self.cmap
1233
1234  def __str__(self):
1235      return "ColorMap(%s)" % self.cmap
1236
1237  def __eq__(self, other):
1238      return self.cmap == other.cmap
1239
1240  def __neq__(self, other):
1241      return self.cmap != other.cmap
1242
1243  def __hash__(self):
1244      return hash(self.cmap)
1245
1246  def __copy__(self):
1247      return ColorMap(self.cmap)
1248
1249  def __deepcopy__(self, memo):
1250      return ColorMap(self.cmap)
1251
1252  def __del__(self):
1253      pass
1254
1255  def __getstate__(self):
1256      return self.cmap
1257
1258  def __setstate__(self, state):
1259      self.cmap = state
1260
1261  def __getattr__(self, attr):
1262      return getattr(self.cmap, attr)
1263
1264  def
```

I DATI RICEVUTI

Una volta ricevuti i dati e le foto dall'ISS, li abbiamo visionati, scartato le foto che non potevano essere utilizzate e analizzato gli indici NDVI.





LA TRACCIA MUSICALE

Il valore dell'NDVI è stato utilizzato per creare una melodia su scala pentatonica (ogni volta che il valore aumenta si alza anche la nota), la latitudine è associata al basso, mentre i dati di pitch e roll sono stati utilizzati per il volume della cassa e del rullante, infine il charleston è generato dall'imbardata sul canale sinistro e dai dati della bussola sul canale destro.

```
1 from osgeo import *
2 from threading import *
3 import csv, pathlib, time
4 import sys, random, math
5 # Import the data file
6 # where starts the program
7 # file=PATH_TO_PATH_FILE with name="data.csv"
8 # data file
9 # data file
10 # data file
11 # data file
12 # data file
13 # data file
14 # data file
15 # data file
16 # data file
17 # data file
18 # data file
19 # data file
20 # data file
21 # data file
22 # data file
23 # data file
24 # data file
25 # data file
26 # data file
27 # data file
28 # data file
29 # data file
30 # data file
31 # data file
32 # data file
33 # data file
34 # data file
35 # data file
36 # data file
37 # data file
38 # data file
39 # data file
40 # data file
41 # data file
42 # data file
43 # data file
44 # data file
45 # data file
46 # data file
47 # data file
48 # data file
49 # data file
50 # data file
51 # data file
52 # data file
53 # data file
54 # data file
55 # data file
56 # data file
57 # data file
58 # data file
59 # data file
60 # data file
61 # data file
62 # data file
63 # data file
64 # data file
65 # data file
66 # data file
67 # data file
68 # data file
69 # data file
70 # data file
71 # data file
72 # data file
73 # data file
74 # data file
75 # data file
76 # data file
77 # data file
78 # data file
79 # data file
80 # data file
81 # data file
82 # data file
83 # data file
84 # data file
85 # data file
86 # data file
87 # data file
88 # data file
89 # data file
90 # data file
91 # data file
92 # data file
93 # data file
94 # data file
95 # data file
96 # data file
97 # data file
98 # data file
99 # data file
100 # data file
```

Find out more on

RISULTATO



Dopo la creazione della traccia musicale, attraverso Google Earth Studio abbiamo ricreato il percorso effettuato dall'ISS, anche grazie ai dati EXIF contenuti nelle immagini.