

Homework - CNN

(인공지능 프로젝트)

학번: 2013311659

이름: 곽 창 근

여러 전처리 과정과 feature들을 결합해서 사용해가며 평점을 예측해보고자 시도하였다. 보고서 설명은 전처리 과정을 먼저 설명하고, 이후 모델에 대해 설명하겠다.

1. 전처리 과정

① H열: Original_language: 원작 언어

i. 사용 계기

원작 언어를 안다면 어느 나라 영화인지, 또는 영화 시장 스케일이나 해당 영화를 보는 관객이 pool이 다르기에 영화 평점에도 영향을 끼칠 것이라 생각하여 해당 feature를 사용하게 되었다.

원작 언어의 경우 train data에 29개 종류의 언어가 있었고, test data의 경우 train data에 있는 언어 중 11개 만이 존재하였다. 그래서 충분히 판단 지표로써 사용할 수 있었다.

ii. 전처리 방법

각각의 언어를 숫자로 매기기엔 부적절 하고, 분류가 적절하다 생각하였기에 one hot 인코딩을 하였다. 29개 종류의 언어가 있었으므로, 각 언어에 대해 0~29까지 번호를 매기고 각 번호에 해당하는 노드만 1로 설정하고 나머지는 0으로 설정해서 one hot 인코딩을 구현하였다. 자세한 코드구현은 "AIproject 전처리.ipynb" 파일의 "사용 언어 one_hot으로" 파트에 코드구현과 간단한 주석을 달아놓았다.

② J열: overview: 줄거리

i. 사용 계기

줄거리는 그 영화의 대략적인 스토리를 알 수 있기에 이를 통해 사람들의 평가를 어느정도 예측할 수 있을 것이라 생각하여서 사용하게 되었다. 가공되지 않은 줄거리 텍스트를 그대로 사용하기엔 부적절하다 생각하여서 자연어 전처리 과정을

거쳐서 사용하였다.

ii. 사용된 핵심 모듈

Overview 전처리 과정에는 사용된 중요 라이브러리가 몇몇 있기에 설명하겠다.

Pandas : csv파일을 불러올 때 사용

BeautifulSoup : html 태그들이 붙어있는 경우 제거해주기 위해 사용

Re : 정규식을 사용하여 치환해줄 때 사용

Nltk

Nltk.corpus의 stopwords : 불용어 처리를 할 때 사용

Nltk.stem.snowball의 SnowballStemmer : 여러 stemmer들 중 SnowballStemmer 사용

iii. 전처리 방법

우선 data파일에 저장된 가공되지 않은 overview를 불러온다. 그 후 1) html 태그가 붙어있다면 제거한다. 2) 정규식을 사용해서 영문자가 아닌 문자를 공백으로 전환하고, 3) 모든 영문자를 소문자로 변환해준다. 4) 불용어를 set로 변환해준 후, 5) 이를 이용하여 불용어를 제거한다. 6) 마지막으로 모든 단어들의 어간을 추출 (stemming)한 후 모두 묶어 공백으로 구분된 문자열로 결합하여 진행하였다.

아래 사진은 자연어 처리 전의 텍스트와 자연어 처리 이후의 텍스트이다. 이를 보면 불용어들이 사라지고, 단어들이 stemming 된 것을 볼 수 있다.

In [205]:	x_train[:3] # overview 잘 들어갔나 확인
Out [205]:	array([[{"Lied by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences."}, {"Cheated on, mistreated and stepped on, the women are holding their breath, waiting for the elusive "good man" to break a string of less-than-stellar lovers. Friends and confidants Vannah, Bernie, Glo and Robin talk it all out, determined to find a better way to breathe.'}, {"Just when George Banks has recovered from his daughter's wedding, he receives the news that she's pregnant ... and that George's wife, Nina, is expecting too. He was planning on selling their home, but that's a plan that -- like George -- will have to change with the arrival of both a grandchild and a kid of his own."}], dtype=object)
In [206]:	clean_overviews[:5]
Out [206]:	['led woodi andi toy live happili room andi birthday bring buzz lightyear onto scene afraid lose place andi heart woodi plot buzz circumst separ buzz woodi owner duo eventu learn put asid differ', 'cheat mistreat step women hold breath wait elus good man break string less stellar lover friend confid vannah berni glo robin talk determ in find better way breath', 'georg bank recov daughter wed receiv news pregnant georg wife nina expect plan sell home plan like georg chang arriv grandchild kid',

그래서 overview는 2가지 형태로 저장하였는데,

하나의 overview를 위의 자연어 처리 과정을 거쳐서 저장해준 것이고,

두번째는 위의 과정을 하고, 중복 단어들을 제거해서 어떤 단어들이 쓰였나를 저장한 단어뭉치를 저장한 것이다.

③ C열: Budget: 예산 & P열: revenue: 수익

i. 사용 계기

영화를 만드는 데 있어서 사용 예산이 더 많이 들었다면 그만큼 더 완성된 영화일 수 있고, 그러면 관람객들의 평점이 더 높을 수 있겠다 생각해서 예산을 사용하게 되었다. 그리고 예산이 많이 들었다 해서 무조건 잘 만든 영화라 판단하기는 어렵고 수익율과 함께 보는게 좋겠다 생각해서 예산과 수익을 활용해서 수익율을 계산하였고, 이를 feature로 주었다.

ii. 전처리 방법

```
def MinMaxScaler(data) :  
    numerator = data - np.min(data, 0)  
    denominator = np.max(data, 0) - np.min(data, 0)  
    # noise term prevents the zero division  
    return numerator / (denominator + 1e-5)
```

위와 같은 MinMaxScaler 함수를 구현해서 정규화하여 주었다.

예산은 값의 범위가 너무 크고 sparse 하기에 MinMaxScaler 함수를 이용해서 0~1 사이 값으로 정규화를 해주었다. 수익율은 예산과 수익을 이용해서 수익율을 계산해보았는데 수익율도 생각보다 범위가 크고 sparse 하기에 역시 정규화를 해주었다.

④ K열: popularity 인기도

i. 사용 계기

인기도는 이름에서부터 평점에 큰 요인을 끼칠 것이라 판단해서 사용하게 되었다.

ii. 전처리 방법

'③'에서와 같은 방식으로 MinMaxScaler를 통해 정규화하여 주었다.

⑤ M열: 영화 제작 회사

i. 사용 계기

실제 영화 흥행에 있어서 영화 제작 회사가 큰 영향을 끼치는 것으로 알고 있다. 그러기에 이를 알면 평점도 예측 가능할 것이라 생각하여서 고려하였다.

ii. 사용 핵심 모듈

영화 제작 회사들이 json파일 형식으로 저장되어 있었다. 그렇기에 json 모듈을 import 하여서 분석하였다.

iii. 전처리 방법

```
def jsonRead(li_input, frequency, str_target):
    # 함수 내의 주석 예는 영화 제작 회사를 추출할 때로 예를 들을
    i = 0
    li_items = []
    u_li_items = [] # 이는 unique한 회사이름 저장하기 위해
    # while문 통해 들면서 회사 추출해서 저장
    while True:
        if i == len(li_input): #반복문이 끝나는 조건 : 끝까지 다 봤을 때
            break

        try:
            j = (li_input[i][0])
            j = j.replace("'", '"') # json.loads 할 때 '로 되었으면 안되고 "로 바꿔야함
            j = json.loads(j)
        except Exception as e:
            #print(e)
            j = li_input[i][0]

        index = 0
        while index != len(j):
```

우선 data에서 긁어온 회사 정보들을 json.loads() 함수에 사용할 수 있게 query문을 수정해준 후, query문의 'name' 정보를 받아서 저장하였다. Query문 수정 및 json query문을 통해 값을 받는 자세한 과정은 위의 jsonRead 함수에 작성되어 있다. 그리고 train data에서 같은 회사가 중복해서 5회 이하로 나온 경우에는 이 회사의 영향력을 평가하기가 힘들고, 중요하지 않은 회사일 것이라 판단하여서 5회 이하로 등장한 회사들은 jsonRead, keywords_frequency 함수를 이용하여 제거하였다. 그 결과 501 개의 회사 종류들을 저장해주었다. 그리고 '①' 에서와 같은 이유로 convertOnehot 함수를 구현하여 one_hot 인코딩을 해주었다.

⑥ D: 장르

i. 전처리 방법

장르도 '⑤'에서 구현했던 방식으로 jsonRead, keywords_frequency, convertOnehot 함수들을 이용해서 진행하였다. 장르도 json query문 형태로 저장되어 있었고, 이를 jsonRead 함수를 통해 query문의 값을 읽어서 저장해주고, convertOnehot 함수를 통해서 one_hot 인코딩을 해서 저장해주었다. 장르는 20여 종의 장르가 나왔기에 one_hot 인코딩 결과 20개 노드로 나왔다.

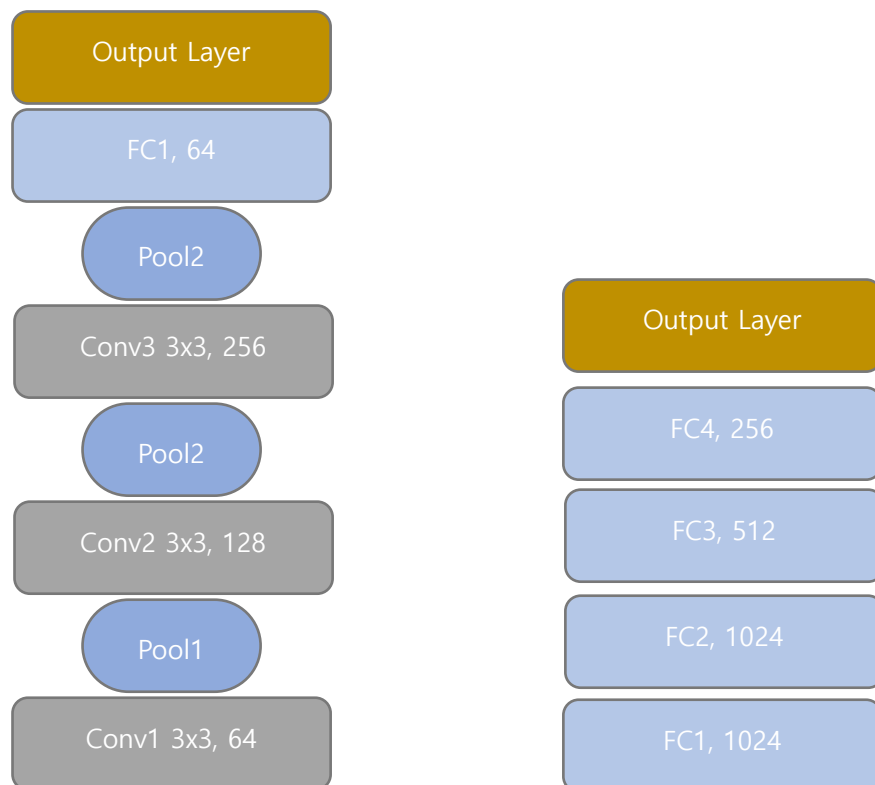
2. 모델

① CNN

포스터 이미지를 이용해서 평점을 예측하는 CNN 모델을 구현해보았다. Batch_size = 100, Iter 횟수 = 10, dropout = 0.75 였다.

구성은 아래 왼쪽과 같이 Conv, Pool을 3번 해주고 FC layer를 거쳐서 점수를 예측하는 모델이다. 층을 더 쌓고 Conv layer도 더 넣어주고 싶었지만 이미지의 크기가 300x200이어서 batch_size가 100이었음에도 불구하고 한정된 GPU 사이즈에서는 아래 왼쪽 모델이 한계였다.

아래 모델로 실험한 결과 포스터마다의 예측값이 4~6 사이에서 모두 정해졌고, 모두 값이 거의 같게 나왔다. 이는 영화 포스터를 가지고 평점을 예측하는 데에는 유의미한 정보를 주지 않았다 결론을 내리게 되었다. 또한 이미지 크기가 300*200*3로 커서 계산 시간도 매우 길어지고 pooling을 통해서 이미지 사이즈를 줄이려고 해도 큰 사이즈여서 Iteration 횟수도 얼마 할 수 없고 layer도 충분히 쌓을 수 없는 단점이 있었다. 그래서 이미지 포스터를 이용하는 CNN 모델은 사용하지 않았다.



② FC (Fully Connected Neural Network)

원작 언어와 예산, 수익율, 인기도, 영화 제작 회사, 장르 등을 이용하여 평점을 예측하는 FC 모델을 제작해보았다. 구현 모델은 위의 오른쪽과 같이 구성하였다. Batch_size = 200, Iter 횟수는 3000, Dropout = 0.75로 하였고, batch_norm_wrapper 함수를 구현하여 batch normalization을 해주었다.

Hidden layer는 총 4개로 구성되어 있고 앞쪽부터 1024, 1024, 512, 256 개의 노드로 구성되어 있다.

이미지 포스터를 이용한 CNN 모델보다는 확실히 평점에 더 영향을 주는 요소들이기에 더 정확하게 판단하는 모델이 나왔다. Epoch를 반복해가며 학습할 때의 cost 변화를 보면 처음엔 큰 값을 갖다가 나중엔 0.1까지도 내려가는 등 학습이 진행되고 있다는 것을 알 수 있었고, CNN에서는 예측평점이 모두 거의 일정했는데 FC 모델에서는 다양한 조건에 따라 다양한 평점을 주어주는 것을 통해서 훨씬 정확하게 판단하는 모델이라 결정하였다. 그래서 최종적으로 이 모델을 선택하게 되었고, train data들을 갖고서 측정해본 RMSE 값은 1.084였다. Test data들의 평점은 주어지지 않았기에 실제 test data들의 RMSE는 알 수 없다.

3. 예측 평점

Test data 999개에 대해서 예측 평점을 소수점 첫째 자리까지 나타낸 값들이다.

0	6.1	26	7.3	52	7.7
1	7.4	27	7	53	7.6
2	7.7	28	7.2	54	6.8
3	6.9	29	6.9	55	6.9
4	7.1	30	6.5	56	6.2
5	7.1	31	6.7	57	6.8
6	6.8	32	6.6	58	6.8
7	7.6	33	6.9	59	6
8	6.3	34	7.5	60	6.1
9	7	35	7.2	61	6.9
10	7.7	36	7.3	62	5.8
11	7.7	37	7	63	7.1
12	6.4	38	8.1	64	6.5
13	6.3	39	7.3	65	7.2
14	6	40	6.8	66	6.8
15	7.6	41	5.8	67	7.5
16	6.7	42	6.8	68	7
17	5.9	43	7	69	8
18	7.2	44	8	70	7.1
19	6.8	45	5.6	71	6.8
20	6.5	46	7.6	72	7.9
21	7.2	47	6.8	73	8.1
22	7.8	48	7.5	74	6.8
23	6.9	49	7.8	75	6.8
24	5.9	50	7.8	76	6.5
25	6.5	51	7.4	77	6.5

78	8.1
79	6.9
80	7.9
81	7.6
82	7.3
83	7.6
84	7.9
85	6
86	6
87	7.4
88	7.3
89	6.1
90	7.4
91	7.5
92	7
93	6.8
94	7.4
95	6.9
96	7.6
97	7.2
98	6.8
99	6
100	7.3
101	8.2
102	7.5
103	7.7
104	7.1
105	7.3
106	6.9
107	7.1
108	7.5
109	7.7
110	7.2
111	7.2
112	5.8
113	5.6
114	7.9
115	6.2

116	7.2
117	7.3
118	7.9
119	6
120	6.8
121	6.6
122	7.4
123	6.5
124	7.6
125	7.8
126	6
127	7.6
128	5.5
129	6.7
130	7.1
131	6.9
132	5.8
133	7.9
134	6.3
135	7
136	6.6
137	7.7
138	7.8
139	6.4
140	5.4
141	7.7
142	6.1
143	7.1
144	7.8
145	7.1
146	5.6
147	7
148	7
149	6.5
150	8
151	8
152	7.2
153	7.2

154	5.9
155	7.7
156	7.7
157	7.6
158	6.9
159	6.7
160	6.8
161	7.8
162	7.7
163	8
164	6.4
165	6.8
166	6.1
167	6.7
168	6.8
169	7.1
170	7.4
171	6.4
172	6.9
173	7.2
174	6.6
175	6.9
176	7.5
177	6.9
178	7.1
179	7.2
180	7.3
181	6.8
182	6.8
183	6.8
184	7.5
185	6.1
186	7.2
187	6.8
188	5.8
189	7.7
190	6.6
191	7.6

192	7.7
193	6.8
194	6.6
195	6.6
196	7.5
197	5.2
198	7
199	8.1
200	7.1
201	6.3
202	6.6
203	7
204	6.5
205	6.7
206	7.8
207	6
208	7.3
209	5.2
210	7.5
211	6.6
212	7.9
213	7.1
214	7.6
215	6.7
216	6.2
217	7.8
218	7.5
219	7.3
220	7.1
221	5.9
222	7
223	7
224	7.5
225	7.1
226	7.7
227	7.8
228	7.4
229	7

230	5
231	6.3
232	6.1
233	7.3
234	7.3
235	6.1
236	7.8
237	6.8
238	7.9
239	7.4
240	6.3
241	6.8
242	7.1
243	7.3
244	6.7
245	6.5
246	6.7
247	7.4
248	5.8
249	7.7
250	6.8
251	5.3
252	6.3
253	7.8
254	6.4
255	6.7
256	7.6
257	7.4
258	6.9
259	6.7
260	7.6
261	8
262	6.4
263	7.6
264	6.2
265	7.6
266	6.8
267	6.8

268	6.9
269	7.5
270	6.9
271	6.6
272	5.1
273	6.8
274	5.1
275	6.9
276	6.7
277	7.1
278	5.9
279	7.8
280	5.4
281	5.6
282	6.9
283	7.9
284	7.8
285	8
286	7.3
287	7.5
288	7.3
289	5.4
290	7.3
291	7
292	7.7
293	7.3
294	6.4
295	6.1
296	5.6
297	5.7
298	6.5
299	6.5
300	6.4
301	8
302	7.7
303	6.8
304	7.6
305	6.6

306	7.8
307	7.9
308	7
309	7.1
310	6.2
311	6.8
312	7.6
313	8
314	5.1
315	7.2
316	7.3
317	6.8
318	7.9
319	6.9
320	5.2
321	5.6
322	4.9
323	7.7
324	7.1
325	6.8
326	7.7
327	7.2
328	7.2
329	6.4
330	6.4
331	5.8
332	6.9
333	7
334	6.8
335	7.7
336	8
337	7.8
338	6.7
339	6.3
340	8
341	6.8
342	7.6
343	7.5

344	7.1
345	7.5
346	7.5
347	6.8
348	7.3
349	6.4
350	6.6
351	7
352	7.2
353	7.4
354	7.4
355	7.2
356	7.8
357	7.1
358	6.3
359	5.9
360	8
361	7.2
362	5.8
363	6.1
364	6.1
365	7.1
366	7.4
367	7.7
368	5.3
369	7.5
370	5.8
371	6.7
372	5.8
373	6.8
374	7.6
375	6.6
376	8.4
377	5.8
378	5.3
379	5.2
380	7.8
381	7.4

382	5.8
383	6.6
384	6.7
385	7.8
386	7.1
387	7.4
388	6
389	5.3
390	6.8
391	6.5
392	6.1
393	6.5
394	7.6
395	7.5
396	7.9
397	5.8
398	7.4
399	7
400	7.3
401	5.3
402	6
403	6.5
404	7.7
405	7.8
406	5.9
407	6.7
408	6.8
409	7.2
410	5.8
411	6.7
412	7.2
413	7.7
414	6.9
415	6.8
416	7.3
417	6.6
418	7.2
419	6.7

420	6
421	7.9
422	5.5
423	6.5
424	6.5
425	4.8
426	5.9
427	7.4
428	6.9
429	7
430	7.2
431	6.8
432	5.2
433	6.8
434	6.2
435	5.5
436	6.8
437	7.3
438	7.3
439	6.6
440	7.5
441	7
442	5.8
443	5.7
444	5.8
445	6.3
446	6.6
447	5.9
448	7.1
449	6.6
450	6.9
451	7.3
452	7
453	7.4
454	7.1
455	7
456	5.3
457	7.6

458	6.8
459	7.2
460	7.2
461	6.6
462	6.3
463	7.1
464	6.7
465	6.6
466	7.4
467	6.8
468	5.2
469	5
470	7.4
471	7.3
472	6.7
473	7.9
474	6.4
475	7.6
476	7.8
477	7.2
478	7.3
479	8
480	6.5
481	6.2
482	7.3
483	6.2
484	7.7
485	7.9
486	7.3
487	7
488	7.6
489	6.1
490	7.7
491	7.7
492	7.3
493	6.2
494	7.1
495	6.7

496	6.7
497	7.6
498	7.1
499	7.9
500	7.7
501	7.6
502	6.3
503	6.8
504	6.6
505	6.1
506	5.7
507	6.7
508	6.5
509	5.3
510	6.9
511	7
512	6.8
513	7.1
514	7.8
515	7.8
516	6.4
517	6.8
518	5.4
519	4.5
520	5.4
521	7.6
522	6.4
523	7.2
524	6.9
525	7
526	5.7
527	7.2
528	6.3
529	7.4
530	7.6
531	6.2
532	7.9
533	6.9

534	5.5
535	6.7
536	5.8
537	7
538	7.4
539	6.7
540	6.5
541	6.7
542	7.2
543	7.5
544	6.7
545	7.4
546	7.9
547	5.8
548	5.9
549	7
550	6.9
551	6.4
552	7.2
553	7
554	5.8
555	6.2
556	6.1
557	7.8
558	7.7
559	7
560	7.5
561	5.3
562	7.7
563	7.5
564	6.6
565	6
566	6
567	6.7
568	5.9
569	6
570	7.7
571	6.3

572	5.4
573	7.9
574	7.5
575	6.2
576	7.6
577	7.8
578	6.8
579	6
580	7.4
581	5
582	5.9
583	7.5
584	6.2
585	7.6
586	7
587	7.2
588	6.1
589	6.9
590	7.5
591	7.4
592	7.2
593	7.5
594	7
595	7
596	7.7
597	6.7
598	7.6
599	7.8
600	5.6
601	7.4
602	5.4
603	7.2
604	7.2
605	5.3
606	7.8
607	6.6
608	6.7
609	6.3

610	6.8
611	7.2
612	7.1
613	6.5
614	7.5
615	6.6
616	6.7
617	7
618	7.2
619	7.7
620	7.3
621	7.5
622	6.4
623	7.5
624	5.1
625	4.9
626	5.3
627	6.4
628	7.5
629	7.1
630	6.7
631	6.8
632	6.2
633	7.4
634	5.4
635	5.7
636	7.6
637	7.3
638	4.5
639	7
640	6.9
641	7
642	6.5
643	7.1
644	6.6
645	7.2
646	7.3
647	6.8

648	6.9
649	6.5
650	7.7
651	7.1
652	7.3
653	7.2
654	7.7
655	7.2
656	7.4
657	6.3
658	6.8
659	6.9
660	7.3
661	5.6
662	6.9
663	6.8
664	7.1
665	7.4
666	7
667	7.8
668	6.7
669	6.2
670	5.5
671	7.2
672	6.2
673	6.6
674	5.6
675	6.5
676	7.5
677	5.2
678	5.1
679	6.8
680	7.3
681	7.8
682	6.1
683	7.6
684	7.3
685	7

686	6.8
687	8
688	6.1
689	7.4
690	6.1
691	6.5
692	6.8
693	6.8
694	7.6
695	7.1
696	7.5
697	6.9
698	5.8
699	6.6
700	7.2
701	7.4
702	7
703	6.9
704	6.2
705	7.9
706	7.4
707	7.2
708	6.5
709	7
710	5.4
711	7.7
712	7
713	5
714	7.2
715	7.3
716	5.5
717	5.7
718	6.2
719	7.1
720	6.6
721	6.7
722	6.3
723	7.3

724	6.4
725	5.2
726	7.5
727	6.7
728	7.4
729	6
730	7.1
731	6.2
732	7.3
733	7.1
734	6.1
735	6.6
736	7.6
737	6.5
738	6.8
739	7.5
740	6.3
741	5.8
742	7.7
743	6.9
744	7.4
745	7.1
746	7.2
747	6.9
748	5.5
749	7.1
750	4.8
751	6.1
752	6.6
753	6.1
754	7.5
755	5.2
756	6.3
757	7.4
758	5.6
759	7
760	4.7
761	7.5

762	7.6
763	6.9
764	7.6
765	6.6
766	5.8
767	6
768	6.1
769	6.7
770	6.8
771	7.9
772	5.5
773	7.3
774	6.7
775	4.9
776	6.3
777	6.7
778	7.8
779	8.1
780	7
781	6.7
782	7.8
783	6
784	5.8
785	7.7
786	7.2
787	7.6
788	6.6
789	7.1
790	5.8
791	6.3
792	7.3
793	6.2
794	8
795	7.1
796	7.4
797	6.3
798	7.3
799	5.7

800	7.5
801	6.8
802	6.5
803	7.3
804	5.7
805	7.9
806	6.8
807	6.5
808	5.8
809	7.7
810	7
811	5.9
812	4.4
813	7.5
814	7.2
815	7.7
816	5.6
817	7.1
818	7.4
819	7
820	6
821	6.7
822	6.7
823	6.8
824	5.4
825	6.6
826	7.1
827	6.1
828	7.6
829	6.2
830	7
831	7.1
832	5.9
833	5.8
834	7.5
835	6.2
836	7.3
837	6.3

838	6.8
839	7.5
840	6.2
841	6.9
842	5.6
843	6.2
844	5.6
845	5.7
846	5
847	6.9
848	7.3
849	5.1
850	6
851	6.4
852	7.2
853	7.4
854	6.5
855	6.3
856	5.5
857	7
858	6.1
859	7.8
860	7.1
861	6.6
862	7.2
863	6.7
864	6.3
865	6.8
866	6.8
867	7
868	5.5
869	6.3
870	8.3
871	6.9
872	7.8
873	5.6
874	6.6
875	6

876	7.1
877	8
878	6.2
879	5.5
880	6.7
881	4.7
882	5.6
883	4.8
884	6.2
885	5.7
886	6.9
887	6.3
888	7.5
889	7.7
890	5.3
891	7.1
892	6.8
893	6.1
894	6.4
895	6.3
896	6.7
897	7.4
898	7.2
899	6.3
900	6.9
901	8.3
902	7.6
903	8
904	6.3
905	7.2
906	7.1
907	7.8
908	5.7
909	7.1
910	6.8
911	6
912	7.9
913	6.3

914	6.4
915	7.6
916	6.6
917	7.7
918	6.8
919	6.9
920	7.4
921	7.4
922	7.4
923	7.7
924	7
925	6.8
926	7.4
927	6.3
928	7.8
929	6.8
930	7.9
931	6.2
932	6.7
933	4.7
934	5.8
935	6.5
936	5.4
937	7.1
938	6.9
939	6.8
940	7.7
941	6.5
942	6.4
943	6.5
944	7
945	6.8
946	7.1
947	7.2
948	5.8
949	7.6
950	6.6
951	6.2

952	6.3
953	5.4
954	4.9
955	6.3
956	7.5
957	4.7
958	7
959	5.7
960	7.6
961	6.1
962	5.3
963	6.1
964	6
965	6
966	5.8
967	4.9
968	7.3
969	6.5
970	6.8
971	6.3
972	5.9
973	6.7
974	7
975	7.1
976	6
977	8
978	7.4
979	6.1
980	7.3
981	7.2
982	6.9
983	3.6
984	6.4
985	6.7
986	7.5
987	7.9
988	6.2
989	5.9

990	7.3
991	7
992	7.2

993	7.9
994	6.4
995	8.2

996	6.5
997	5.8
998	6.8

4.