Executive Summary:

Netflix wants to address user decision paralysis by minimizing the average browsing time. To achieve this goal, we conducted experiments manipulating four factors: Tile Size, Match Score, Preview Length, and Preview Type. Our experimental journey consisted of running a general 2^k factorial test followed up by narrowing down the optimal range of each factor further and further. Eventually, we landed upon our optimal condition at {Tile Size = 75, Match Score = 75, Tile Size = 2, Preview Type = “TT”} which resulted in a mean average browsing time of 10.08 minutes.

Introduction:

Users of the popular streaming service Netflix often face decision paralysis due to the sheer number of options available on what to watch. This can result in the undesirable scenario of users becoming overwhelmed by the number of options and resultantly choosing simply not to watch anything.

To overcome this problem, Netflix wants to minimize the average browsing time a user takes before picking something to watch. In this quest to minimize the average browsing time, we manipulated four different factors: Tile Size, Match Score, Preview Length and Preview Type.

In the process of our experimentation we firstly performed some 2^k tests to help get a general idea of which factors were significant and what general values we might want to aim towards.

From there, we specifically chose conditions to help narrow down our search. With each set of new conditions tested, we narrowed down our range of possible values further and further. At each step of this process, we performed global F tests and one sided T tests to help ensure that the differences we were finding were indeed statistically significant.

Eventually by our last test, we were testing an extremely small range of values. From these conditions, we would choose our optimum.

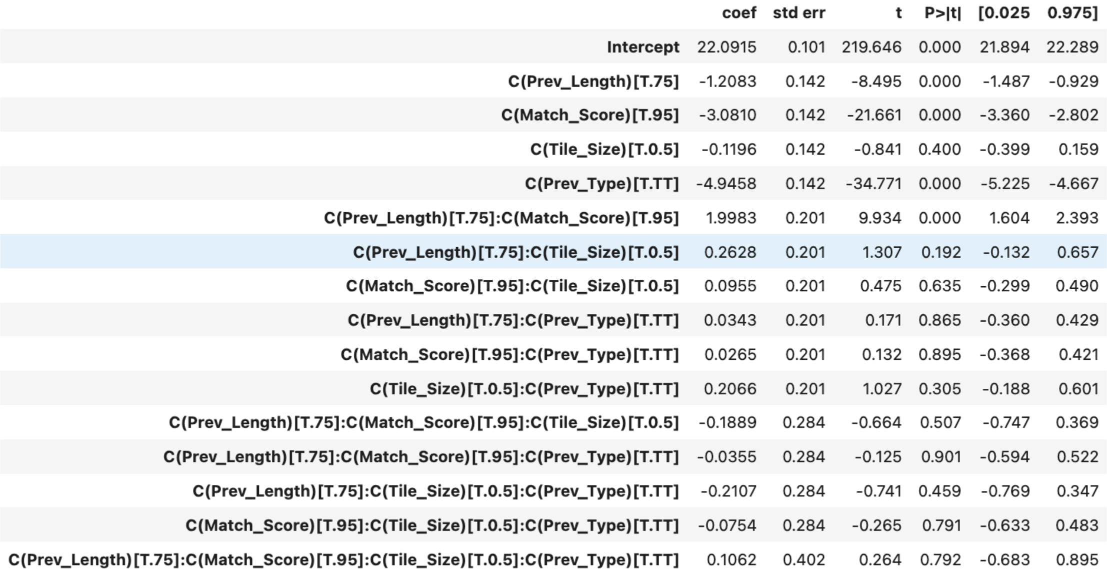
The Experiments:

To try to minimize the browse time, we initiated an exploration of the significance of all factors and their interactions. To do this we performed a two-level factor screening experiment where k = 4 (the number of factors) and chose some initial levels to test.

Factor Name	**Levels**
Prev. Length	40, 75
Prev. Type	TT, AC
Match.Score	50, 95
Tile.Size	0.2, 0.5

Reasons for selecting the levels: First, we designated one level with the default value. Following that, we chose another value significantly different from the default, aiming to narrow down the range within which the optimal point may be situated.

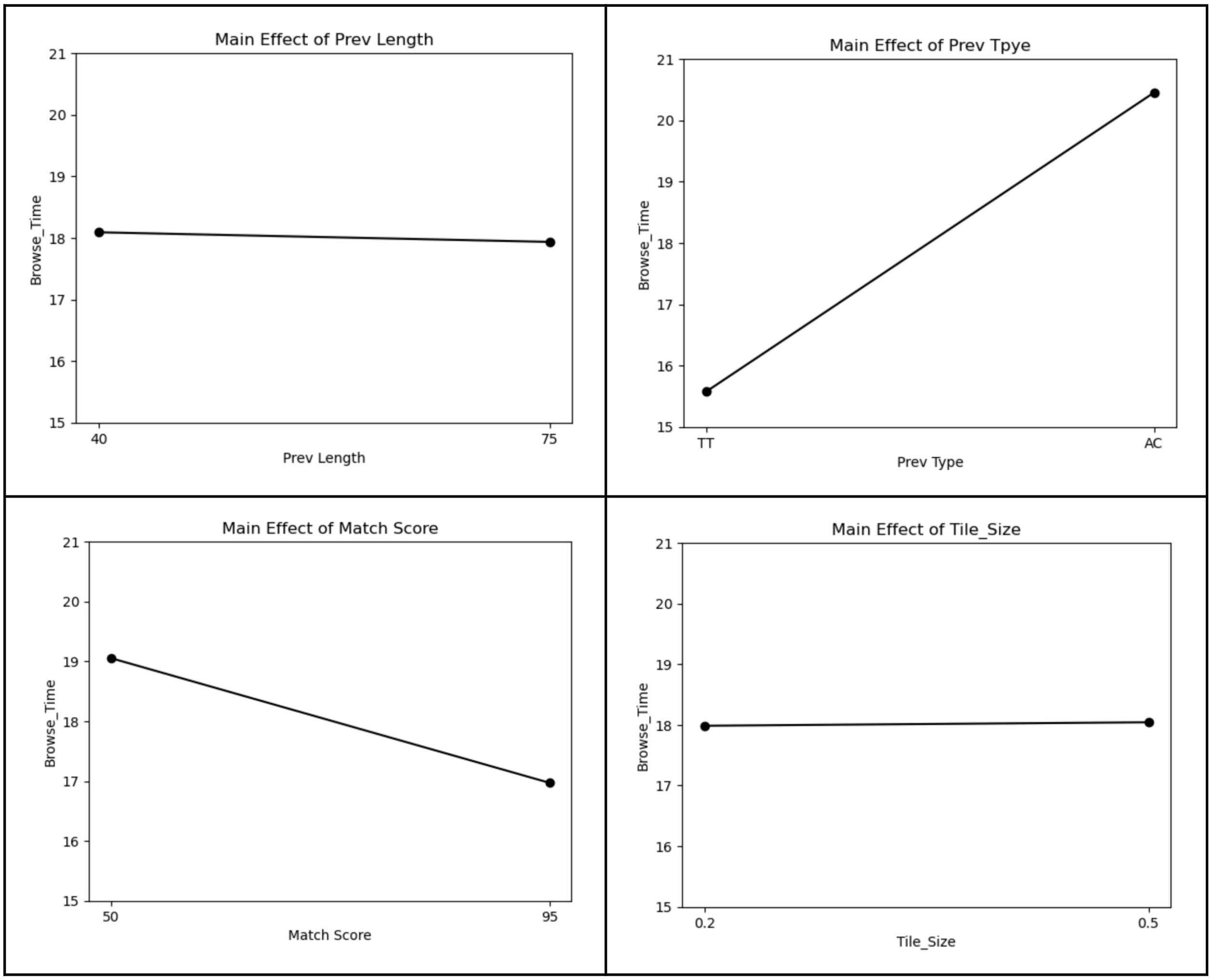
Regression Output



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	22.0915	0.101	219.646	0.000	21.894	22.289
C(Prev_Length)[T.75]	-1.2083	0.142	-8.495	0.000	-1.487	-0.929
C(Match_Score)[T.95]	-3.0810	0.142	-21.661	0.000	-3.360	-2.802
C(Tile_Size)[T.0.5]	-0.1196	0.142	-0.841	0.400	-0.399	0.159
C(Prev_Type)[T.TT]	-4.9458	0.142	-34.771	0.000	-5.225	-4.667
C(Prev_Length)[T.75]:C(Match_Score)[T.95]	1.9983	0.201	9.934	0.000	1.604	2.393
C(Prev_Length)[T.75]:C(Tile_Size)[T.0.5]	0.2628	0.201	1.307	0.192	-0.132	0.657
C(Match_Score)[T.95]:C(Tile_Size)[T.0.5]	0.0955	0.201	0.475	0.635	-0.299	0.490
C(Prev_Length)[T.75]:C(Prev_Type)[T.TT]	0.0343	0.201	0.171	0.865	-0.360	0.429
C(Match_Score)[T.95]:C(Prev_Type)[T.TT]	0.0265	0.201	0.132	0.895	-0.368	0.421
C(Tile_Size)[T.0.5]:C(Prev_Type)[T.TT]	0.2066	0.201	1.027	0.305	-0.188	0.601
C(Prev_Length)[T.75]:C(Match_Score)[T.95]:C(Tile_Size)[T.0.5]	-0.1889	0.284	-0.664	0.507	-0.747	0.369
C(Prev_Length)[T.75]:C(Match_Score)[T.95]:C(Prev_Type)[T.TT]	-0.0355	0.284	-0.125	0.901	-0.594	0.522
C(Prev_Length)[T.75]:C(Tile_Size)[T.0.5]:C(Prev_Type)[T.TT]	-0.2107	0.284	-0.741	0.459	-0.769	0.347
C(Match_Score)[T.95]:C(Tile_Size)[T.0.5]:C(Prev_Type)[T.TT]	-0.0754	0.284	-0.265	0.791	-0.633	0.483
C(Prev_Length)[T.75]:C(Match_Score)[T.95]:C(Tile_Size)[T.0.5]:C(Prev_Type)[T.TT]	0.1062	0.402	0.264	0.792	-0.683	0.895

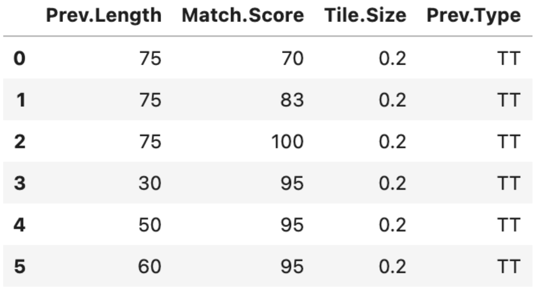
According to the output table, we concluded that Prev. Length, Prev. Type and Match.Score have significant main effects, while Tile.Size is not a significant factor. Prev. Length and Match Score is the only significant interaction.

Main Effect Plots



Based on the main effect plots, it is evident that TT is a more favorable choice than AC in terms of minimizing browsing time. A match score of 95 is superior to 50, and a preview length of 75 is marginally better than 40. To confirm our findings we performed a likelihood ratio test where: H_o: $B_3 = B_{13} = B_{14} = B_{23} = B_{24} = B_{34} = B_{123} = B_{124} = B_{234} = B_{134} = B_{1234} = 0$, wiith B_1 = preview length, B_2 = match score, B_3 = tile size, B_4 = preview type, and interactions represented by all beta interactions(B_{123} represents interaction between B_1,B_2,B_3). By comparing the full model to our reduced model we got a p-value of .95 indicating that the factors in our null hypothesis are not significantly different from 0. After we found that tile size was not significant in determining browse time, we decided to keep the default value of .2 going forward in all of our data collection. Next we wanted to test if there was a significant difference between preview type(TT or AC). We performed a t-test with H_o: $u_1 = u_2$, Ha: $u_1 != u_2$ where u_1,u_2 are the mean browsing times of TT and AC respectively. We ran this on the data collected from the 2^k test so that there are 800 units in each condition. After confirming equal variances through a t-test with a large p-value, we performed the test and got a p-value of 0.00 so we can reject the null hypothesis and have evidence that the mean of TT is different from AC. After looking at the means of the two conditions, it looks like TT has a lower mean on average than AC. So we conducted another t-test this time with H_o: u_1 >= u_2, H_a: u_1 < u_2. This yielded a p-value of 0.00 so again we can reject the null and have evidence for the alternative where TT has a lower mean browse time than AC. This agrees with our likelihood ratio test!

Now we know we only want to test tile size at .2 and preview type at TT, we collected a wide array of values. The thought was to submit small chunks for experimentation, check the means, and learn which values are “best”. For preview length we tested [30,50,60] using default values for everything else and then tested [70,83,90] for match score. We wanted to test values that were significantly different from each other and different from our initial 2k test.

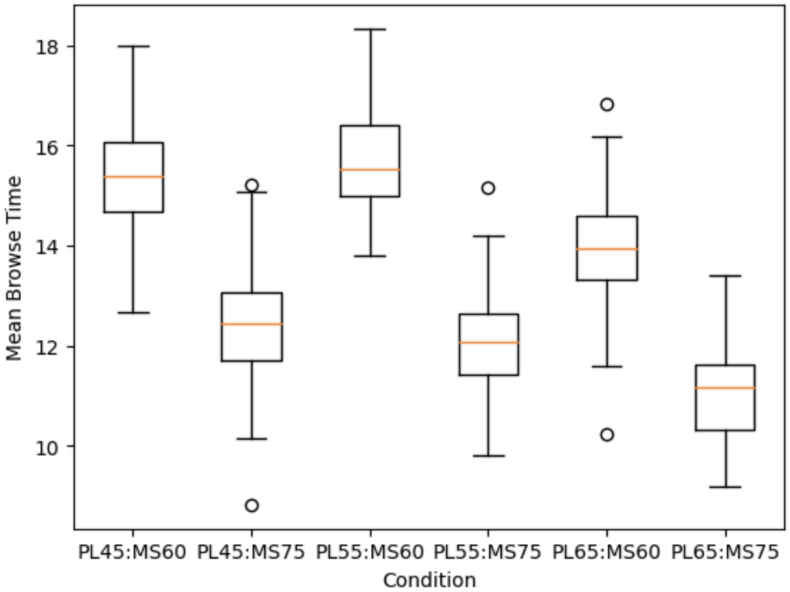


	Prev.Length	Match.Score	Tile.Size	Prev.Type
0	75	70	0.2	TT
1	75	83	0.2	TT
2	75	100	0.2	TT
3	30	95	0.2	TT
4	50	95	0.2	TT
5	60	95	0.2	TT

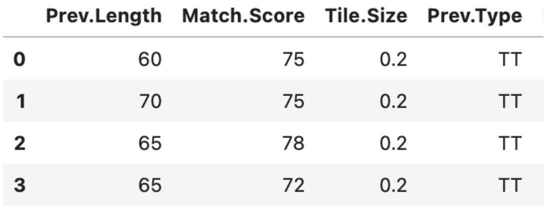
Following the same procedure as in the preview type test, we conducted pairwise comparisons using a t-test. The best combination was preview length: 75, match score: 70, tile size: .2, and preview type: TT, which resulted in a mean browse time of 10.27. We found no statistical difference between preview length 50 and 60 which were both better than 30. Match score 70 was significantly better than both 83 and 100 match score.

Next we wanted to try a range of values for preview length against similar match scores to our best. So our next data collection was preview lengths of [45,55,65] against match scores of [60,75]. We avoided using 70 again as we wanted to try a wider range of values before focusing on a specific number. The conditions with match score 75 were much better than that of 60 with the best being preview length 65, match score 75. However, this resulted in a mean of 11.06

which is close but worse than our previous round of data collections best.



Putting all the data collection we had done together, it looked like match scores of 75 were ideal and preview lengths around 65. We wanted to fill in the “gaps” and submitted just 4 more tests pictured below.



	Prev.Length	Match.Score	Tile.Size	Prev.Type
0	60	75	0.2	TT
1	70	75	0.2	TT
2	65	78	0.2	TT
3	65	72	0.2	TT

This time, preview length of 70 and match score of 75 yielded the best mean browse time of 10.39. Again, it is still worse than our previous best but very close.

For the next round of data collection we chose to test all preview lengths at 70 against match scores of [70,71,73,77]. This is because we had just gotten a good result using preview length 70 and match score 75 so wanted to test lots of values around that. We also threw in our previous best(preview length 75, match score 70) to see if it still compared and to get a sense of how much noise the data was generating. Again, we conducted pairwise comparisons using a t-test. The result was that preview length 75, match score 70 was still the best but the others were very close. With only three tests left before hitting our cutoff of 40 tests, we decided to take a couple “dart throws” around our best score. This meant testing the preview lengths of 75 against match scores of [68,73,75]. We settled on a preview length of 75 over 70 as the last round of testing showed it was slightly better and we already had tested most of the match scores associated with it. Then we chose match scores around 70. As it turns out, the smallest mean browsing time we obtained was when preview length: 75, match score: 75, tile size: .2, and preview type: TT. The mean browse time was 10.08.

Conclusion:

The location of the minimum browse time when viewing netflix suggestions was when the preview length = 75, match score = 75, tile size = .2, and preview type = TT. This resulted in a browse time of just 10.08 minutes with the 95% confidence interval being (9.89, 10.26). The main limitation to our findings is that we performed multiple tests on the same data which could result in multiple comparison problems. This would give us a higher chance of committing a type 1 error where we incorrectly reject the null when it was actually true. While we could have run a grid search to find the global minimum, we saved time and resources by only running 40 experiments. We may have missed the true global minimum but hopefully came very close. Lastly because we ended up testing one condition twice, there is an issue of differing sample size as we have more information on that one experiment than all the others.