Work 2: Neural Networks

Computational Intelligence

Nikita Belooussov

Barcelona, December 18, 2021

1 Definition of the input and output linguistic variables

For the project the variables can be organized into three different groups. These categories are input, input/output, and output. The variables and which category they belong to can be seen below.

- Input: audience, time slot, duration, and program rank category
- Input/Output: interest of the channel
- Output: Price

The reason for this type of separation is due to the some inputs being inputs that function only as inputs and not as outputs for other controllers. The *interest of the channel* variable is both an output and an input. It will not be given, but calculated using the *audience*, *time slot*, and *duration*. The *interest of the channel* will then be used with the *program rank category* to determine the *price*. The lab requires the for each of the inputs to be represented by trapezoidal membership functions. In order to do this, research needs to be done for the range of each of the inputs possible. The research will be done for US markets and will try to reflect these numbers.

1.1 Audience

It is important to note that in this project audience and viewership are interchangeable. An article published by Variety[1] has a list of the top 100 shows in USA with the number of viewers for each show. While this is may not be the best representation of all the shows possible, due to there being a much larger amount of shows that may have smaller viewership, especially on smaller networks such as *Syfy*, this will provide a range that is reasonable for and will only require minimal amount of changes to fix. Saying this the range for the shows is between 6,216,000 and 842,000. These numbers also represent the amount of viewers between the ages of 18 and 49. A further look into the list seems like the list it is seen that the difference between the top 3 shows and the rest is very large, with the difference between each top show being at about 1,000,000 and then having about 2,000,000 drop between third and fourth place. As a result, it seemed to be better to set off the top 3 shows in their own membership function. Once these were separated, the rest of the end points were put into more of an equidistant range. This was done due to the program rank already being an input. If the shows amount of viewers was to be divided by the rank, then that would be reinforcing the value that the rank input would be giving. If the rank was not given, then it would be a better option to try to position the end points around some predetermined ranks. The end points for the memberships can be seen in Table 1. The points correlate to the points seen in Fig 1.

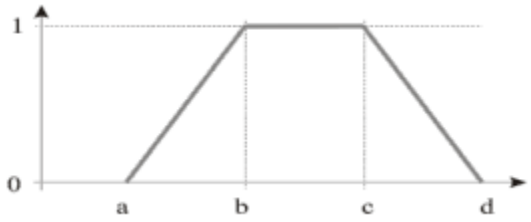


Figure 1: Trapazoid Membership Function

Table 1: Audience Membership Function Endpoints(by million)

Points	a	b	c	d
Low	0	0	1.2	1.3
Medium	1.2	1.3	1.7	1.8
High	1.7	1.8	2.2	2.3
Very High	2.2	2.3	3.25	3.5
High Outliers	3.25	3.5	329	329

In the Table 1, it is seen that the outliers were sectioned off. The start of the group after the outliers also has higher boundaries than normal, this is in order to account for shows that may perform better but are not quite as high as the outliers. After these these two groups the membership functions become to change at the same pattern. Until the lowest viewed shows. In this case, the last membership functions goes to zero, so that it is still able to function even if the show is receiving low views. Again it is important to note that this was created for the top 100 shows in USA, so for a more general model, it may be necessary to either add more membership functions or change the endpoints, depending on how well tuned everything should be.

1.2 Time Slot

The membership functions for time slots will have a range of 24 to make it easier to fit to each hour of the day. The day will be further divided into 5 basic phases which can be seen below:

- Night AM: Hours between midnight and when people usually wake up
- Morning: Before Work
- Day: Hours people generally work during
- Evening: Hours between Work and sleep
- Night PM: Hours between going to sleep and midnight

The way that the membership functions were created around these definitions can be seen in Table 2

Table 2: Time Slot Membership Function Endpoints

Points	a	b	c	d
Night AM	0	0	5	7
Morning	5	7	8	10
Day	8	10	16	18
Evening	16	18	21	22
Night PM	21	22	24	24

These were mainly decided by a general idea of the times when people do things, and not a lot of research was done into determining them. Again these were made with trapezoidal membership functions so the points are the same as in Figure 1.

1.3 Duration

The next input is the duration of the ads. There was a little research done on this and it became apparent that commercial lengths can vary between 15 seconds to 2 minutes. This is of course not considering infomercial, which can be a lot longer. These were not included due to them generally not being placed in the commercials during a show. The duration was broken up into three groups that appeared to be common lengths for commercials, including old formats[2]. In Table 3, one can see how this was translated into end points for the membership functions.

Table 3: Duration Membership Function Endpoints(seconds)

Points	a	b	c	d
15 seconds	0	0	17	25
30 seconds	17	25	40	45
60 seconds	40	45	120	120

It is important to note that any values that are longer than 2 minutes can be changed to two minutes to have the system to continue working. If longer commercials start becoming more common or are more common in the area that this system is being used, then more membership functions can be added to account for the longer format.

1.4 Program rank category

The last true input covered will be the program rank category. For this project it is assumed that the rank means the exact rank that the show has compared to other shows. In this case, it was decided to have 5 different membership functions. The values were

decided by way that favorite lists being numbered usually. Generally when a favorite list is created it is top 10,25,50, and 100. This seemed like a reasonable way to create the membership functions, with each function containing this range, and the last being below 100. Following this structure the membership endpoints can be seen in Table 4

Table 4: Program Rank Membership Function Endpoints

Points	a	b	c	d
10	0	0	8	12
25	8	12	23	27
50	23	27	50	60
100	50	60	110	120
Below 100	110	120	200	200

In Table 4, the way that the membership functions were designed was to have the main parts of the same ranking shows to obtain the same score, but still have a slight improvement if the show is closer to a better category, and a slightly worse ranking if the ranking is closer to a worse category. This effect can be controlled through the a and d end points. This is true for all of the different membership functions and not just for this case.

1.5 Interest of the Channel

This variable is at the same time an output of a fuzzy logic controller and an input to one. It is possible to have the membership functions to not be matching between the two fuzzy logic controllers, but in this case they will be. This will make the relation between the fuzzy logic controller easier to understand and less error prone to when analyzing what is happening. As a result, both the output and input case can be seen in Table 5

Table 5: Interest Membership Function Endpoints

Points	a	b	c	d
A	88	92	100	100
B	78	82	88	92
C	68	72	78	82
D	58	62	68	72
F	0	0	58	62

For this variable, it was decided that the membership functions could help resemble the American letter grade system. This allows for a quick understand of the score of the *interest*, and provides a good representation of how well the *interest* score is doing.

1.6 Price

The *price* is the final output that is expected from the system. This was harder to define, due to the price range of ads varying greatly depending on where it is being shown, what time, and for what program. For example, an ad for the Super Bowl may cost several million dollars, while an ad in the middle of the day would cost only a couple of hundred dollars. It was decided to use the range found on from this article[2]. The reasoning for this was that the values seem to be realistic, but even if they are wrong, it would be easy to adjust them so that it would match up with the correct outputs. The *price* memberships endpoints that were decided can be seen in 6

Table 6: Price Membership Function Endpoints

Points	a	b	c	d
low	0	0	500	750
low-med	250	750	1050	1500
medium	500	1400	1700	2500
high	1500	1950	2250	2750
high	2250	2500	3000	3000

In the end, it the model ended up having five membership functions. This was so that it would be easier to set up the rules. Since both of the inputs had five membership functions, it will be easier to match everything up. If the system needs to be more accurate then more membership functions can be added as needed.

2 Define the rules for the two blocks

For this lab a simple rule structure was set up. This was mainly due to the fact that there were so many membership functions. As a result, going through all of the possible combinations would take a very long time, and would not be necessary, since the fuzzy logic controller will essentially do the same operations with the simpler rules defined. If there was a need for more complex rules such as that if *Rank* was high and *Interest* was high that the output should be a little less, this could be done by adding another membership function, but none of these sort of interactions between two channels seemed to be necessary. The rules defined for the *interest of the channel* can be seen in Table 7

Table 7: Interest Rules

Audience	Time Slot	Duration	Interest
Low	None	None	F
Medium	None	None	D
High	None	None	C
Very High	None	None	B
Outlier	None	None	A
None	Night AM	None	F
None	Morning	None	B
None	Day	None	C
None	Evening	None	A
None	Night PM	None	B
None	None	15	F
None	None	30	C
None	None	60	A

The reasoning behind the rules was pretty simple. For the *audience* input, as the value went up so did the *interest*. The process behind the *time slot* input was slightly more complex. This was looking at the time of day and considering how many people would be able to watch TV. It is obvious that during the Night AM hours, that most people will be sleeping and no one will be watching TV. The most people would be watching after work. By using this type of reasoning the rest of the inputs were filled in. Lastly, the duration was also filled in fairly simply. Again, the longer the ad the more it would cost, which was a simple method to follow and to justify.

The rule set for the *price* follows a similar structure. Again with one input and one output to make it easier to keep track of what rules have been used, and since using combinations does not have a large effect on the controller for this problem. The rule set can be seen in Table 8

Table 8: Price Rules

Ranking	Interest	Price
10	None	high
25	None	high-med
50	None	medium
100	None	low-med
Bottom	None	low
None	A	high
None	B	high-med
None	C	medium
None	D	low-medium
None	F	low

For both inputs, the output was simple to determine. In the case of the *ranking*, as the ranking got worse, the price got lower. For the *interest* it was the same, as the grade got

worse, so did the price. As mentioned previously, the rules set up were fairly simple. This was due to there not being a requirement for adding a complex rule. It could be possible to add "weights" to the outputs by having that if the *ranking* is high and the *interest* is an F than to have the Price to be high-med. This did not seem to be necessary for either system. This would be done as fine tuning the the overall system to make sure it works better and closer to expected results.

3 Implement the fuzzy expert system using Matlab.

See attached Matlab files

4 Define three different test cases and execute the system.

Even though only three test cases are asked for, this report will be taking several tests, just to see the effects of the different variables more clearly. The list of the first tests can be seen in Table 9. These tests were used to confirm that the system is operating as intended.

Table 9: Test Scenarios and Results

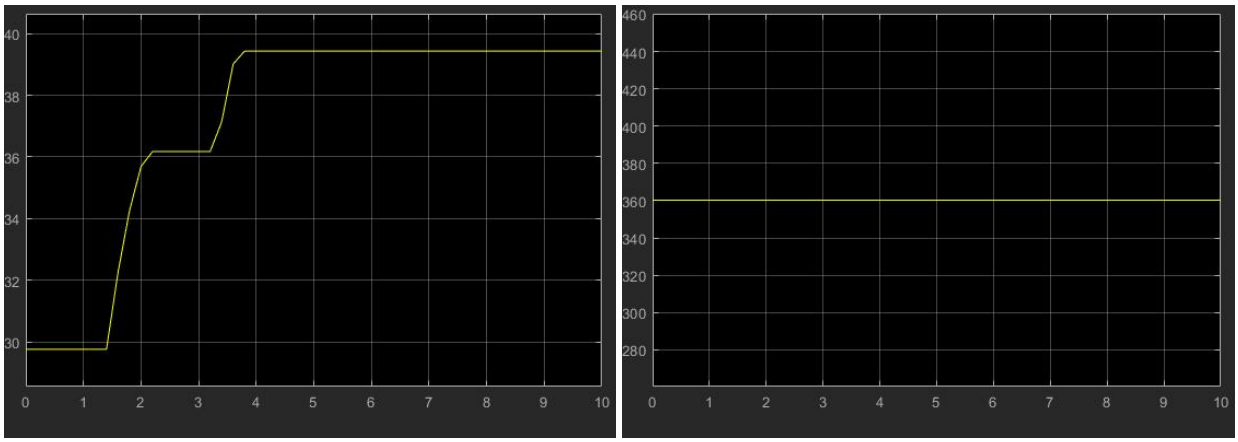
Audience	Duration	Time Slot	Program Rank	Interest Result	Price Result
1	15	1	150	29.76	360
6	60	19	1	95.19	2691
2	30	14	40	75.00	1519
6	15	1	150	39.44	360.2
1	60	1	150	39.44	360.2
1	15	19	150	39.44	360.2
1	15	1	1	29.76	1624

The first three tests in the Table 9 are mainly to confirm that everything is acting the way that it was intended. Since if all the inputs are on there minimums, then the the output should be the minimum. This is seen to be matched in all three of the inputs. That as all of the inputs increase the output also increases. This shows that everything seems to be connected properly and the rules are created correctly. The next four tests were to better confirm this and to see how the inputs affect the final output. From these four tests it is seen that the *audience*, *duration*, and *time slot* have the same affect for the *interest* results and the *price* results. This is because the none of the inputs were weighted in anyway for the calculations. The *program rank* does have a much heavier impact on the *price* than the the other three inputs. This is because essentially each of those inputs attributes to 1/6th of how much the *price* will change, while the *program rank* contributes to 3/6th. As a result it being at a max value raises the *price* a lot more than the other inputs. Overall, the results matched the expected results. Due to how the system was implemented the values did not change from the ones seen in the graph, as a result plots

are not necessary, since they are just a straight line. These tests confirm that the system designed works as intended.

A better representation can be seen with taking one of the inputs and attaching it to a function that slowly increases. After several runs it becomes apparent that with the current set up, even though changing *audience*, *duration*, or *time slot* individually is not enough to cause a change in *price*. As a result, all three have to be changed in order to see *price* changing. The *program rank* input on the other hand, was able to change the *price* on its own. It is not surprising that the *program rank* was able to change. This will all be further discussed further in the project.

The first thing that is worth looking at is how the interest output and price output responded just to the *duration* input changing. In this test as time increase so did the duration input. This can be seen in Figure 2

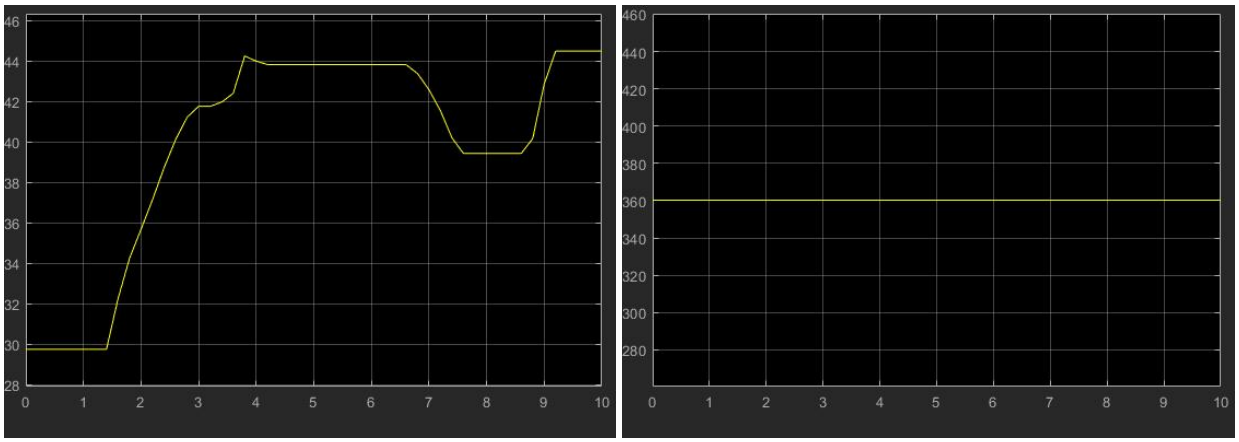


(a) Duration Interest Output (b) Duration Price Output

Figure 2: Outputs for Duration Change

As mentioned previously, the *price* output, seen in Figure 2b, does not change as the duration changes. The reason for why is seen in Figure 2a. While the *duration* does change the *interest* output, it does not reach the value required to enter the next membership function of the price controller. As a result, the *price* output remains to be the same, due to nothing changing from the fuzzy logic's perspective. The plot in Figure 2a, also shows that the logic controller is working as intended with the *interest* increasing as the *ad length* increases. That being said it may be better to try and make the transitions between the plateau longer. This would be achieved by increasing the distance between the endpoint a and endpoint b, and then also increasing the distance between endpoint c and endpoint d.

Next the results of changing two of the inputs to the interest controller will be discussed. The two inputs changed were the *ad duration* and *time slot*. This was done to see if changing two inputs for the interest controller would have an affect on the price. The results for both controllers can be seen in Figure 3

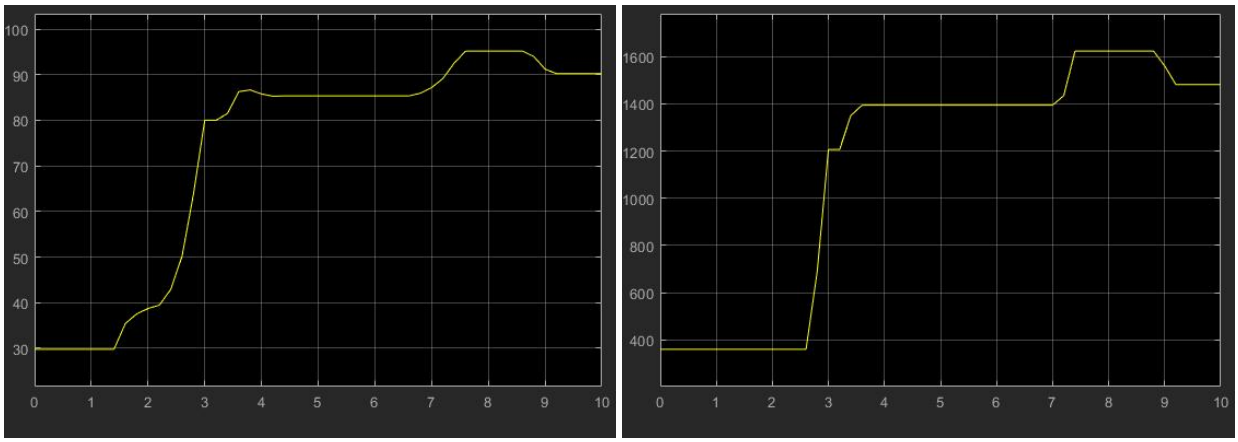


(a) Time and Duration Interest Output (b) Time and Duration Price Output

Figure 3: Outputs for Time and Duration Change

As seen in Figure 3b, the *price* again does not change with both of the values changing, this again is due to the fact that the *interest* output does not reach a high enough value to enter a new membership in the price controller. Again, looking at Figure **??** seems to follow what was expected of the model. The output gradually becomes larger as the day goes on and the audience grows. The large dip seen near the end of the plot is unexpected. This would be near where the *time* function shifts into the *Night PM* membership function, which should produce a lower output than the *Evening* membership function before it. The reason it goes back up may be due to the *Duration* function changing membership functions and outputting a higher value.

Next in the report, the outputs for when all three inputs for the interest controller were changing. This finally produced a change in the *price* values, due to the *interest* value finally being able to get into a new membership function. The results for both of the controllers is seen in Figure 4.



(a) Time and Duration Interest Output (b) Time and Duration Price Output

Figure 4: Outputs for Time and Duration and Audience Change

As seen in 4, the *price* output does not start changing until the *interest* value reaches the value of the new function. This happens at around the 3 second mark when the interest goes above 58. The overall shape of the plot also shows that all of the inputs are working the way that they were expected.The *price* slowly increase as all of the values go up, which is a general trend for all of the variables. The time when it dips is when the *Time Slot* function switches from *Evening* to *Night PM*. This is also an expected result.

The last input that needs to be discussed is how the *rank* affects the *price*. As mentioned before, due to it being a direct input to the price controller, it has a much stronger influence over the *price*. This is confirmed by the plots seen in Figure 5.

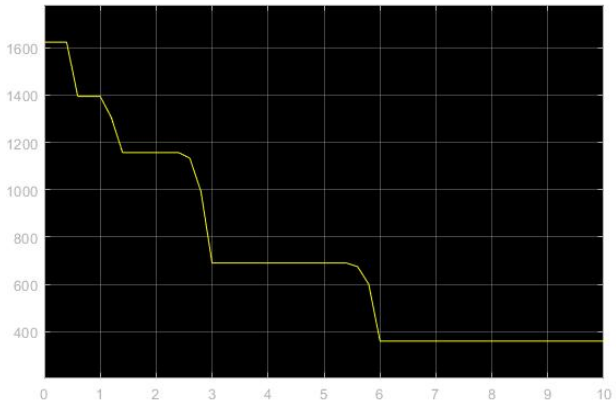


Figure 5: Price Output for Rank

As seen in Figure 5, the rank of the show is able to change the price of the ad. The progress of the *price* values is expected, since it is dropping as the show is becoming less

popular. This is demonstrated through the show's increasing rank. Again, like in the duration plot, it may be better to make the periods of transition longer and shorten the plateaus, but overall this demonstrates that the Price controller is working as intended.

Lastly, it is important to compare the four results together. The comparison can be seen in Figure 6. This is important to see how much influence each input has over the final *price* output.

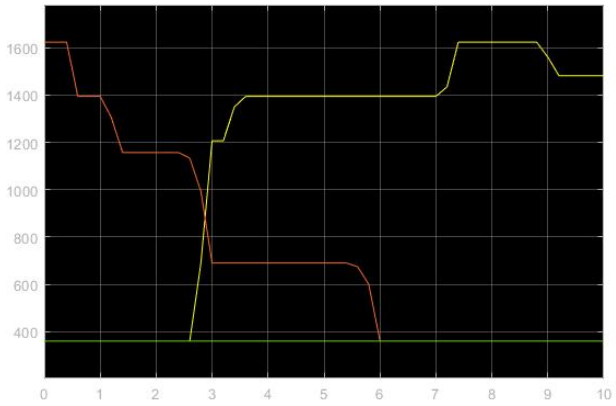


Figure 6: Comparison of the Four Tests

As seen the figure shows, both the final *interest* value and *rank* value play approximately the same role in determining the price of the ad. The *interest* values is represented by the yellow line, and the *rank* is represented by the red line. This is to be expected due to how the system was set up, each having half of the influence. It also shows that the minimum price for all of the tests is the same, which is important, just to make sure there is not something weird happening with the rules. The influence of two or one inputs to the interest clearly do not make any influence on the price as discussed earlier and can be seen in the green line.

Overall, it is evident that the system designed works. There are a few places here and there that can be changed to obtain a better result of the price, but these are relatively small changes, and less important than proving that the concept works. In the future the improvements can be getting a more accurate price range, and adjust so that the transitions between the membership functions to be smoother. The start of the membership functions for the price to start changing for the interest value can also be lowered. In conclusion, the system designed in this report proves the concept works, although there can be minor adjustments to the membership functions to make it work better.

5 Answer the following questions

It is important to not that even though this part is listed after the test cases were set up and their results discussed, these questions were answered before the the simulink model was implemented.

1. What is the influence of the duration over the channel interest if audience is high and program rank category is low?

The *program rank* does not need to be taken into account, since it is not an input of the *interest* output. It also doesn't really matter that the *audience* is high. The affect that the *duration* will have is that the score will go up or down, independent of what the other variables are set to. If the *audience* is high and the *time slot* is set to the best setting, then the *interest* output will be near the max, if the *duration* is max, and will slowly decrease as the duration starts to decrease. If the *audience* is low and the *time slot* is at the lowest, and the *audience* is at the lowest, then the *interest* value will also be the minimum of the range, and start to increase as the *duration* starts to increase. Since there are less membership functions in *duration* then in the other two membership that control *interest*, it is important to note that the changes seen in it are likely to be more rapid and not as smooth. This can be adjusted by making making the distance between a and b , and c and$d$ to be larger. This will make the changes to the output be more subtle.

2. What changes should you make to the system if you want to

a) Apply a discount to the final price of emission depending on the number of previous contracts done with the same company?

Another input to the interest fuzzy logic controller or the price fuzzy logic controller would be able to produce this effect. It would likely be better to put it with the price fuzzy logic controller, so that the interest result could be used for other calculations, without having to worry how the previous contracts would affect it. It would also have a more direct impact on the price, since it isn't a step removed from the overall affect that we desire and it is not being averaged with the other inputs for the interest calculations. As mentioned, it would be possible to lower the price with having the input to the interest fuzzy logic controller, but this is not recommended. The rules would be that as the number of contracts is higher the output should be lower. Another method could be to either add another controller with inputs dependent on the company such as how long and how many ads that determines the discount, and then apply the discount directly to the price that was determined.

b) Consider also the quality of the actors that appear on the advertisement?

The best method for this seems to create another fuzzy logic controller to calculate how much money will be made or lost using the better actors. The controller can take in the inputs like how many, length of ad, and average quality. By doing this the production cost can be calculated and added onto the price of the ad. It is possible to add this to either add this as an input to the price fuzzy logic controller, or just add it directly to the price output. This fuzzy logic controller can also be used by the company buying the add to try to see if they will make money due to quality of the actors improving or lose money. Although this would likely be placed into a separate system. Lastly, it would be possible to create a very simple addition of the quality of actors to the interest or price logic controllers. This depends on what the company would feel better describes the output that it would be changing. Either way, to rules would be set up that as the quality goes up so does the price. This would only be from the point of view of production, and not how the quality of actors helps make money from the ad.

References

[1] Michael Schneider. *100 Most-Watched TV Shows of 2020-21: Winners and Losers*. May 2021. URL: https://variety.com/2021/tv/news/most-popular-tv-shows-highest-rated-2020-2021-season-1234980743/.

[2] Tadmin. *The Average Running Length of Effective TV Commercials - TVA Media Group*. Aug. 2020. URL: https://tvamediagroup.com/average-running-length-effective-tv-commercials/.