1. What has been done?

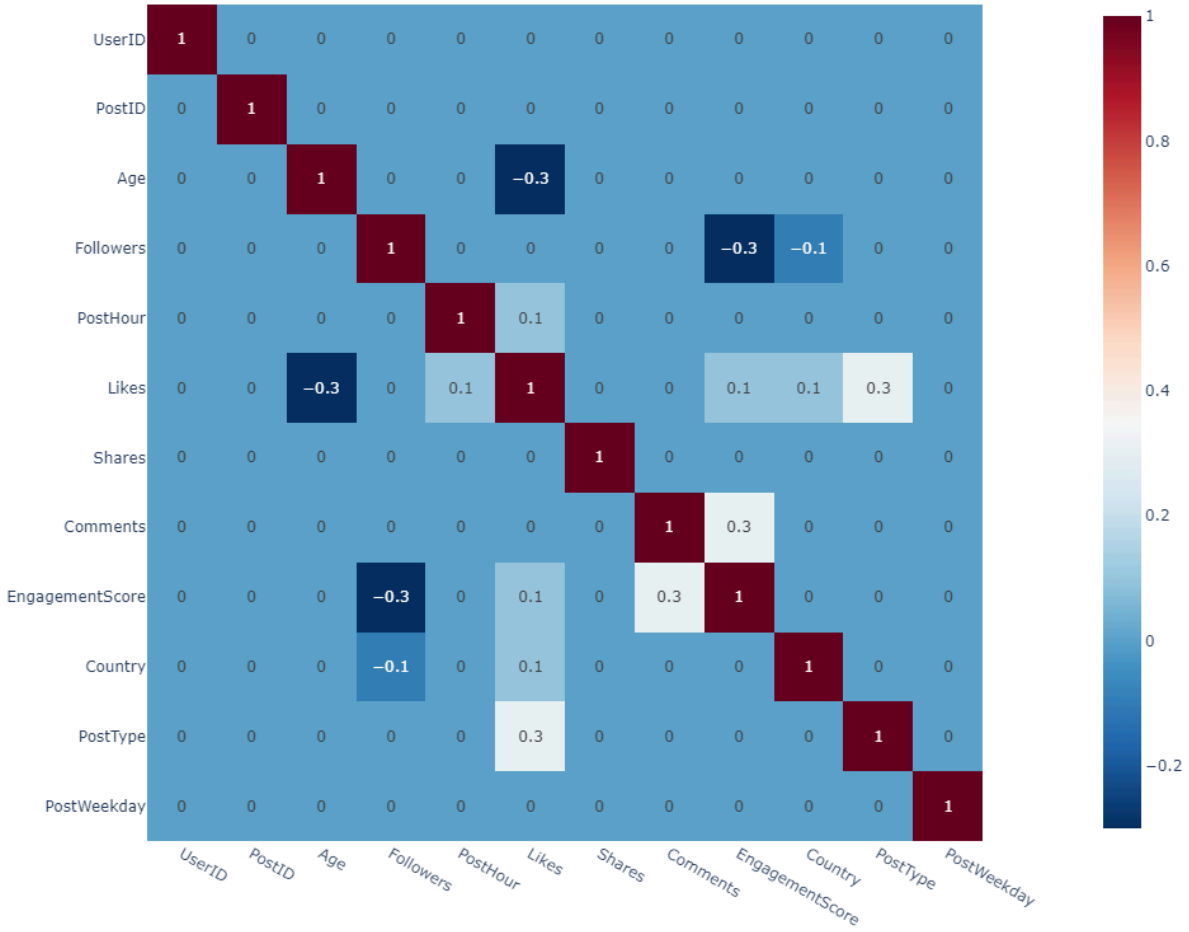
I performed exploratory analysis, transformed the database in numeric format, checked correlations, based on them applied regression models to check feature importance, after I applied trees to make better prediction models. While working with ML models I shuffled metrics and parameters, applied normalization and feature engineering but because the results were the same I used a basic model.

2. Understand the Algorithm: Determine how the EngagementScore is calculated based on various user and post features. Explore what factors influence this score.

From my analysis I find out that the most relevant features that influence EngagementScore are:

- Number of **Followers**
- Amount of **Likes**
- Number of **Comments**

As we can see from correlation matrix other features don't have any correlation with **EngagementScore**:



Also we can see that **EngagementScore** has very light correlation with **Likes** (0.1) metric and moderate correlation with number of **Followers** (-0.3) and **Comments** (0.3). Those are the main metrics we could use to predict EngagementScore of the post.

3. Predict the Unpredictable : Assess if it's possible to accurately predict the EngagementScore for newly created posts.

My assessment is that it is not possible to accurately predict the EngagementScore for new posts due to weak correlation between features we have and the score itself.

I tried to create models and see how close we can get in predicting EngagementScore.

Metric selection:

- UserID and PostID I dropped from the start because those are technical fields that are used in the backend and don't contain any important information.
- Also we can see that Shares field don`t have any correlations with any other so I also dropped that feature from the start.
- After testing all other features that don't have any correlations with EngagementScore those metrics were dropped.

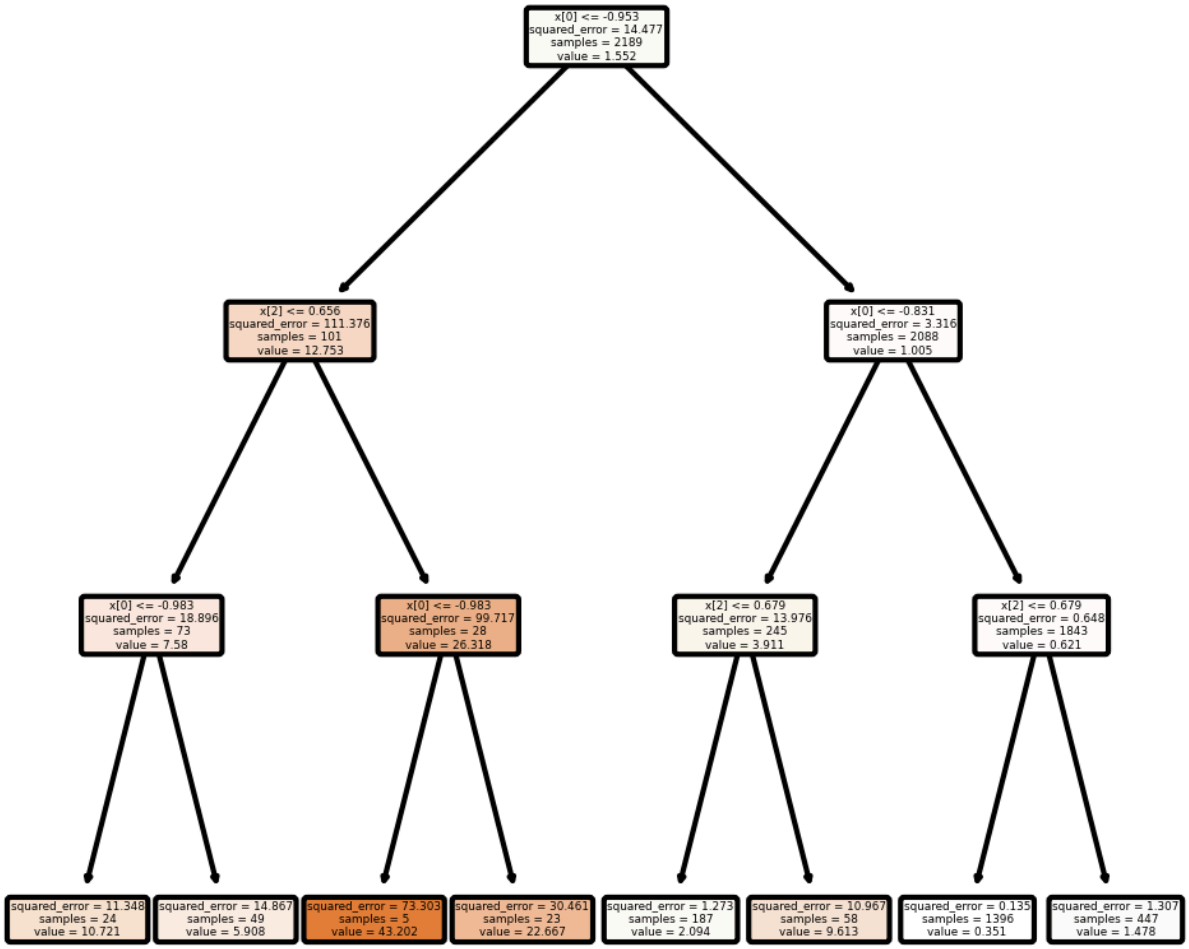
Model selection:
I tested all models from sklearn library - the best performing model was RandomForesrRegressor. After modification we get such results:
R2 train: 0.9044
R2 test: 0.8096

MSE train: 1.6666
MSE test: 3.7407

MAE train: 0.5407
MAE test: 0.5469

After all this is a poor performing model because MAE is 0.54 while average EngagementScore is 1.54.

Plot of our tree:


x[0] <= -0.953
squared_error = 14.477
samples = 2189
value = 1.552
x[2] <= 0.656
squared_error = 111.376
samples = 101
value = 12.753
x[0] <= -0.831
squared_error = 3.316
samples = 2088
value = 1.005
x[0] <= -0.983
squared_error = 18.896
samples = 73
value = 7.58
x[0] <= -0.983
squared_error = 99.717
samples = 28
value = 26.318
x[2] <= 0.679
squared_error = 13.976
samples = 245
value = 3.911
x[2] <= 0.679
squared_error = 0.648
samples = 1843
value = 0.621
squared_error = 11.348
samples = 24
value = 10.721
squared_error = 14.867
samples = 49
value = 5.908
squared_error = 73.303
samples = 5
value = 43.202
squared_error = 30.461
samples = 23
value = 22.667
squared_error = 1.273
samples = 187
value = 2.094
squared_error = 10.967
samples = 58
value = 9.613
squared_error = 0.135
samples = 1396
value = 0.351
squared_error = 1.307
samples = 447
value = 1.478