VIDEO ENGAGEMENT SCORE PREDICTION

Problem:

Before starting the project, it is important to understand the business use case. We might want to ask certain questions to get a clarity on the purpose of this project. Some queries to put forward are:

- What exactly is the business objective?
- Will this model be fed to another machine learning pipeline?
- How does the company expect to use and benefit from the model?

Answers to these will help us in model and its performance metrics selection.

In this problem the objective is to create a model to predict the engagement score of a video. The performance metrics to be used is R2 score.

The provided dataset has the below parameters:

- row_id - Unique identifier of the row
- user_id - Unique identifier of the user
- category_id - Category of the video
- video_id - Unique identifier of the video
- age - Age of the user
- gender - Gender of the user (Male and Female)
- profession - Profession of the user (Student, Working Professional, Other)
- followers - No. of users following a particular category
- views - Total views of the videos present in the particular category
- engagement_score - Engagement score of the video for a user (Target Variable)

Solution:

We know that this is a regression problem as we have a label or target variable to predict. The dependent features are of different datatypes. We have 2 categorical and 6 numerical features (barring the row_id as it is just for the identification purpose and won’t be used for model building).

Approach:

We have followed the below steps:

1. Loading and understanding the data
2. EDA
3. Feature Engineering
4. Feature Selection
5. Base Model

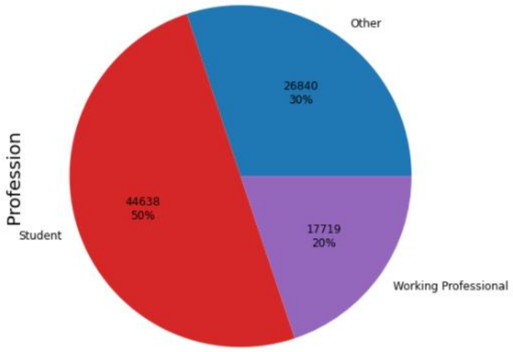
6. Model Comparison
7. Conclusions
8. Next Steps

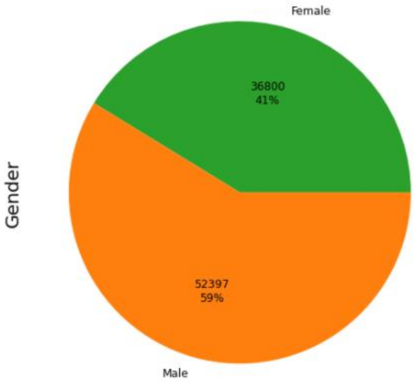
We have imported the required libraries as and when needed

1. Load Data: As a very first step we have loaded the data using pandas read_csv. In this step we get a broad understanding of the train and test dataset. We were able to answer certain questions like the number of rows and columns in both the set, whether any missing values exist and datatypes of all the features.

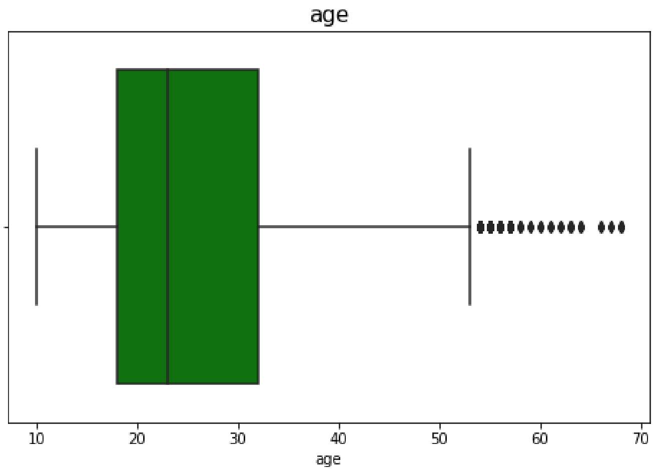
2. EDA: Under exploratory data analysis we explored each of the features closely. Categorical and numerical features were combined and both the groups were investigated separately. We performed both univariate and bivariate analysis to understand the spread and distribution of the data within a feature and also the correlation with other independent features and also with the target. We have used matplotlib and seaborn in abundance for visual interpretations. We drew a lot of important conclusions from the analysis. Key findings:

   Distribution of data within categorical features

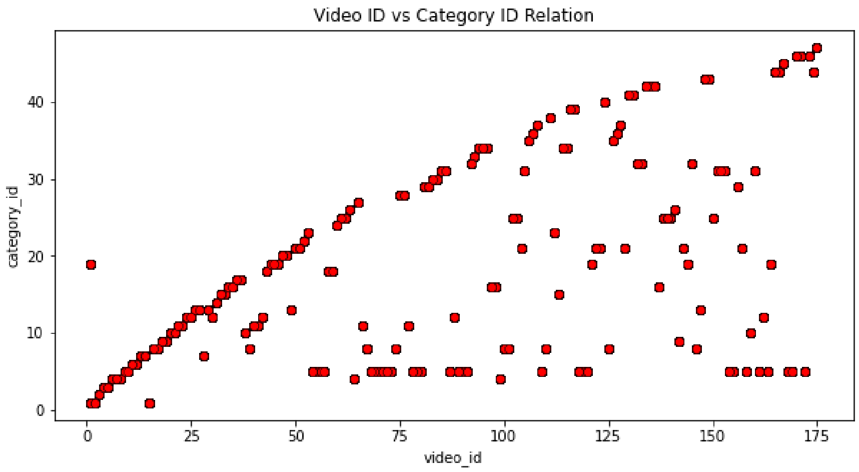




   Outliers in the features

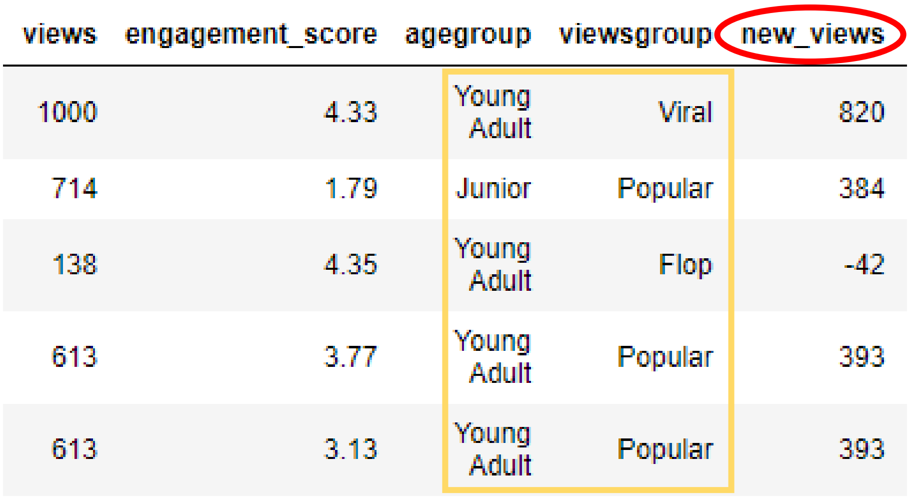


High correlation between video_id and catgory_id



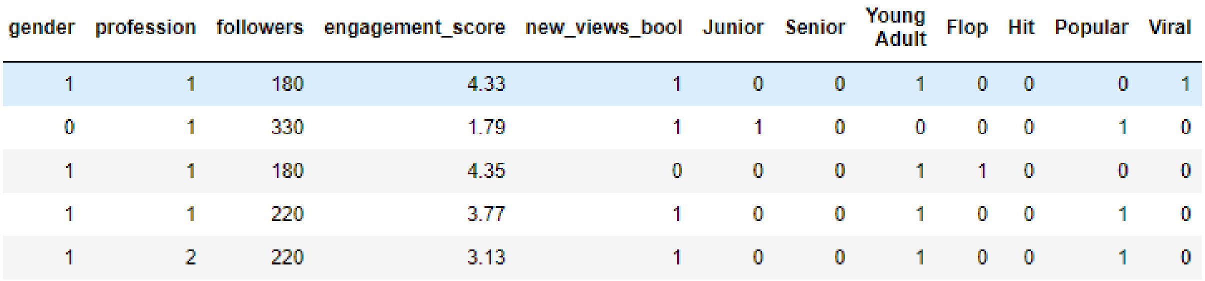
3. Feature Engineering: From the conclusions drawn in the previous section, we have performed different operations in the feature engineering step. Started with handling outliers for age feature followed by grouping the values of age and views into different categories. The latter is called binning (categorizing the values in different bins). We also extracted a new feature, new_views and binned that as well. Binning converted our numerical variables to categorical so we encoded all the non-numerical features using one-hot (features with more than 3 groups) and label encoding techniques.

Illustration of binning and feature extraction:



views	engagement_score	agegroup	viewsgroup	new_views
1000	4.33	Young Adult	Viral	820
714	1.79	Junior	Popular	384
138	4.35	Young Adult	Flop	-42
613	3.77	Young Adult	Popular	393
613	3.13	Young Adult	Popular	393

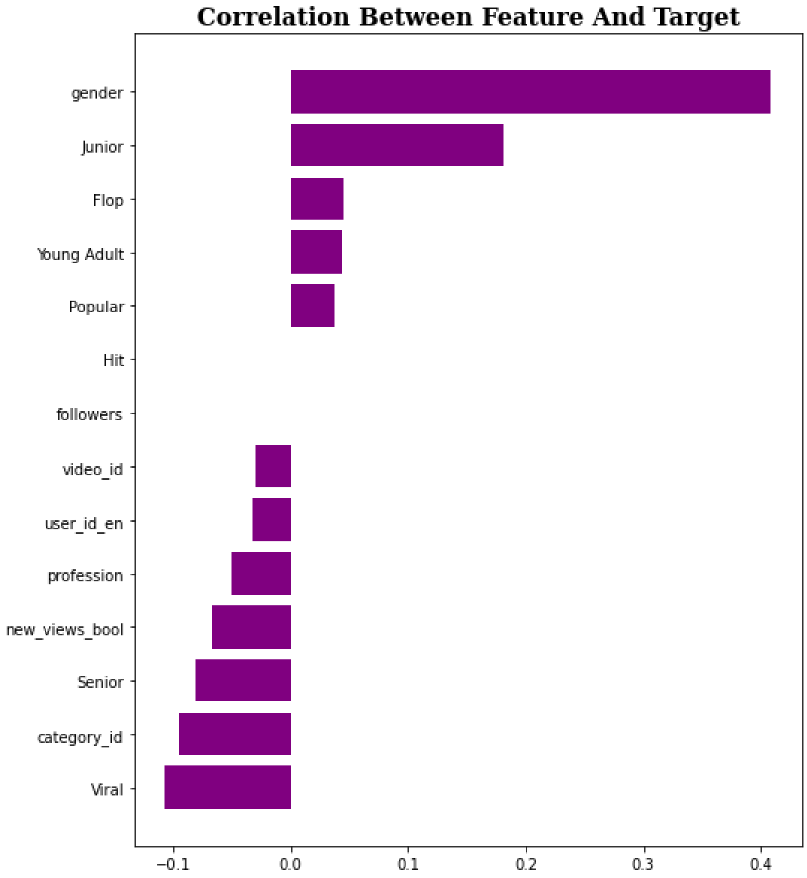
After encoding:



gender	profession	followers	engagement_score	new_views_bool	Junior	Senior	Young Adult	Flop	Hit	Popular	Viral
1	1	180	4.33	1	0	0	1	0	0	0	1
0	1	330	1.79	1	1	0	0	0	0	1	0
1	1	180	4.35	0	0	0	1	1	0	0	0
1	1	220	3.77	1	0	0	1	0	0	1	0
1	2	220	3.13	1	0	0	1	0	0	1	0

Scaling is an integral part of feature engineering process. It is done to scale down the features, which means if the features have different units the values may vary largely and to bring them to the same scale there are methods like Normalization and Standardization.

4. Feature Selection: It is basically checking the relevance of the features in target value prediction. Keeping in mind that one hot encoding and feature extraction have given rise to a number of new features, it is important to ensure that we are not feeding our model with thrash. Here, we have dropped the columns below a threshold value.



5. Base Model: Before starting off with our base model we divided our train data into train and test sets to evaluate our model's performance. We have taken the simple linear regression algorithm for our base model.

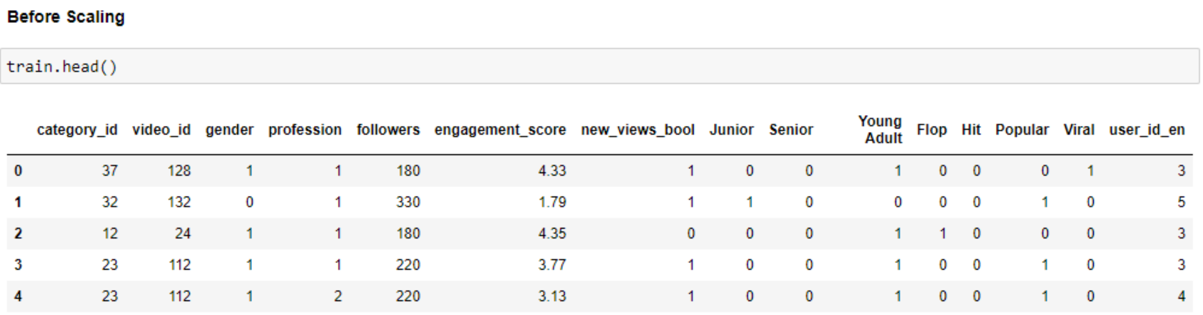
6. Model Comparison: To improve the performance of our model and to ensure that we select the best one, we have taken into consideration 2 ensemble techniques and compared their performance by trying out different hyperparameter tuning techniques.

7. Conclusions: The dataset was not very messy; the best part was that there were no missing data. Some important features which would have been useful:
    a. Date and Time of video upload
    b. Likes and Dislikes on the video
    c. Comments and Ratings enabled or disabled

d. Number of Shares
e. Duration of the video
f. Genre

Experiments and their outcomes:

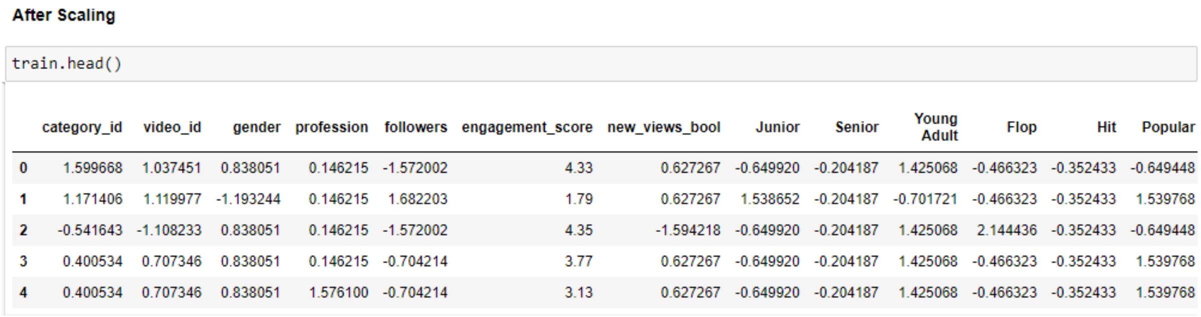
- Scaling did not help in improving the model performance. As we see that the feature values did not vary much and all the columns had values within a small range hence, scaling was not effective in our case:



Before Scaling

```
train.head()
```

	category_id	video_id	gender	profession	followers	engagement_score	new_views_bool	Junior	Senior	Young Adult	Flop	Hit	Popular	Viral	user_id_en
0	37	128	1	1	180	4.33	1	0	0	1	0	0	0	1	3
1	32	132	0	1	330	1.79	1	1	0	0	0	0	1	0	5
2	12	24	1	1	180	4.35	0	0	0	1	1	0	0	0	3
3	23	112	1	1	220	3.77	1	0	0	1	0	0	1	0	3
4	23	112	1	2	220	3.13	1	0	0	1	0	0	1	0	4

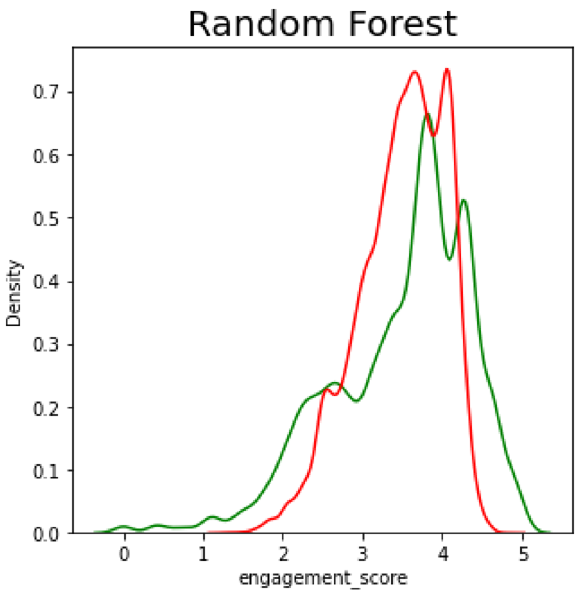


After Scaling

```
train.head()
```

	category_id	video_id	gender	profession	followers	engagement_score	new_views_bool	Junior	Senior	Young Adult	Flop	Hit	Popular
0	1.599668	1.037451	0.838051	0.146215	-1.572002	4.33	0.627267	-0.649920	-0.204187	1.425068	-0.466323	-0.352433	-0.649448
1	1.171406	1.119977	-1.193244	0.146215	1.682203	1.79	0.627267	1.538652	-0.204187	-0.701721	-0.466323	-0.352433	1.539768
2	-0.541643	-1.108233	0.838051	0.146215	-1.572002	4.35	-1.594218	-0.649920	-0.204187	1.425068	2.144436	-0.352433	-0.649448
3	0.400534	0.707346	0.838051	0.146215	-0.704214	3.77	0.627267	-0.649920	-0.204187	1.425068	-0.466323	-0.352433	1.539768
4	0.400534	0.707346	0.838051	1.576100	-0.704214	3.13	0.627267	-0.649920	-0.204187	1.425068	-0.466323	-0.352433	1.539768

- Linear Regression performed poorly. Even without hyperparameter tuning Random Forest and CatBoost did much better than Linear Regression
- Dropping the irrelevant features during feature selection step did improve random forest model's performance
- Removing the video_id column which is highly correlated with category_id degraded the model's performance negligibly.
- The Random Forest Model overfitted the train set as its score on seen data was 0.39 whereas on test set it was just0.34



- We selected CatBoost as our final model as its performance was marginally better from Random Forest

8. Next Steps:

    a. We can try extracting other features and see whether they are helping in predicting the score

    b. We can experiment further with the values of the hyperparameters and see if we see any major difference

    c. R-square is not a recommended performance metrics as it increases with increase in number of features giving us a false picture of our model. It is not always true that higher R-square value means a better model. A very good alternative would be adjuster R-square. Perhaps we can use this metric to judge our models.