**Submitted by : Shubham Sharma**

**Company : Reliance(JIO)**

**Current Designation : Manager – Data Science**

**Email : shubhamsharma1318@gmail.com**

**Problem Statement**

ABC is an online content sharing platform that enables users to create, upload and share the content in the form of videos. It includes videos from different genres like entertainment, education, sports, technology and so on. The maximum duration of video is 10 minutes.

Users can like, comment and share the videos on the platform.

Based on the user’s interaction with the videos, engagement score is assigned to the video with respect to each user. Engagement score defines how engaging the content of the video is.

Understanding the engagement score of the video improves the user’s interaction with the platform. It defines the type of content that is appealing to the user and engages the larger audience.

**Objective**

The main objective of the problem is to develop the machine learning approach to predict the engagement score of the video on the user level.

**Data Dictionary**

You are provided with 3 files - train.csv, test.csv and sample\_submission.csv

**Training set**

train.csv contains the user and video information along with the engagement score

|  |  |
| --- | --- |
| **Variable** | **Description** |
| 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 |

**Evaluation metric**

The evaluation metric for this hackathon is [r2 score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html).

**Guidelines for Final Submission**

Please ensure that your final submission includes the following:

Solution file containing the predictions for the row\_id in the test set (Format is given in sample\_submission.csv)

A zipped file containing code & approach (Note that both code and approach document are mandatory for shortlisting)

Code: Clean code with comments on each part

Approach: Please share your approach to solve the problem (doc/ppt/pdf format). It should cover the following topics:

A brief on the approach used to solve the problem.

Which Data-preprocessing / Feature Engineering ideas really worked? How did you discover them?

What does your final model look like? How did you reach it?

**Approach :**

I had divided the overall problem statement into five parts

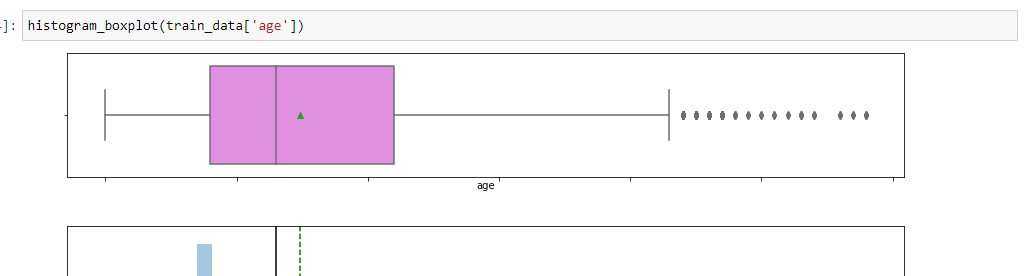
1. Data Exploration and EDA
2. Data Cleaning and feature engineering
3. ML model selection & development
4. Model Validation
5. Prediction for test data
6. Deploy model using flask and docker(Not asked so TBD on demand)

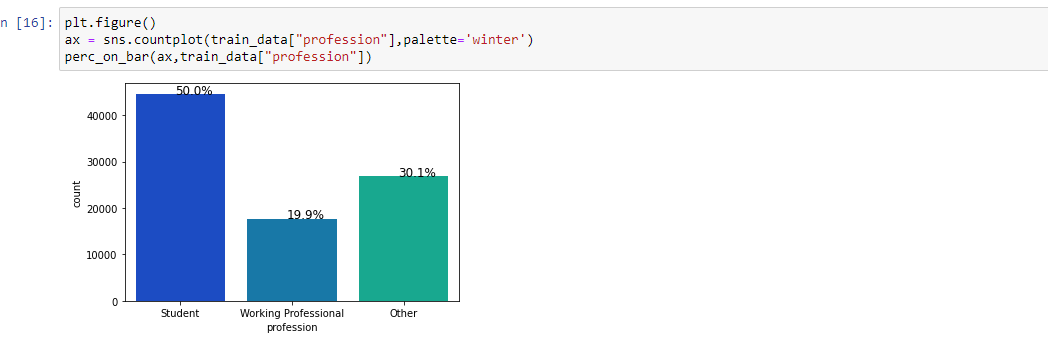
Observation:

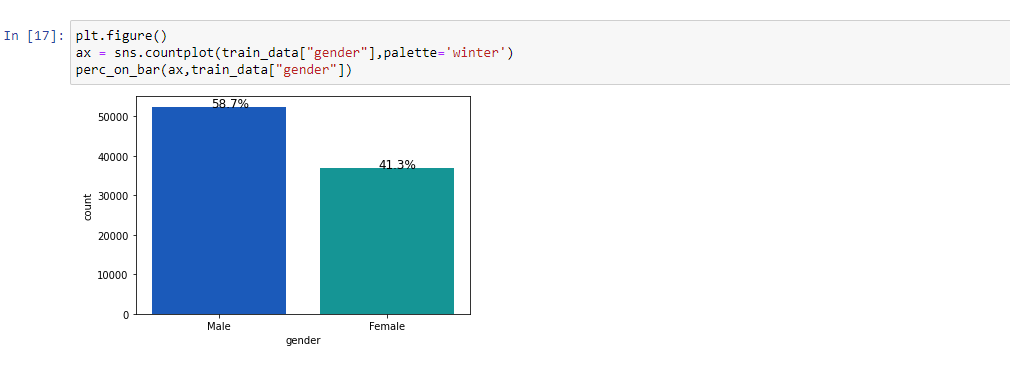
1. **Data Exploration and EDA:**

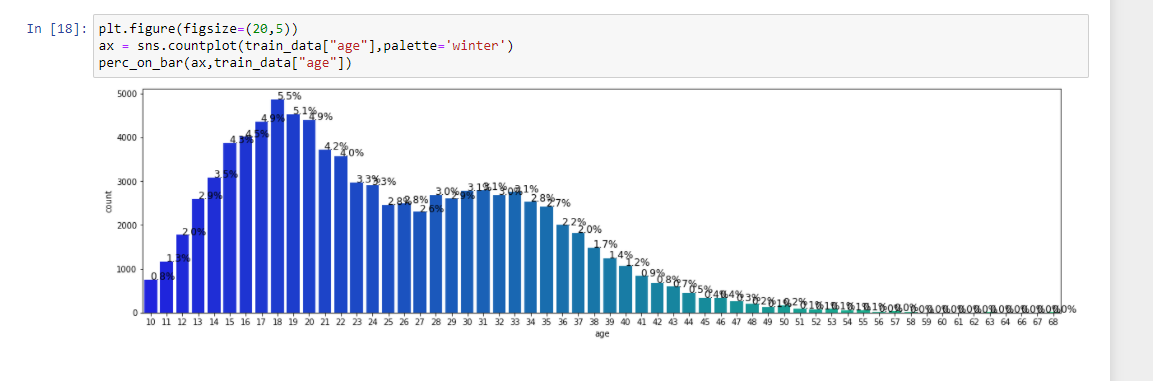
* There are total 10 columns into the training data set
* Most of the user has engagement score between 3 to 5 in train data
* Most of the users of our video website is in the age range of 15 to 40

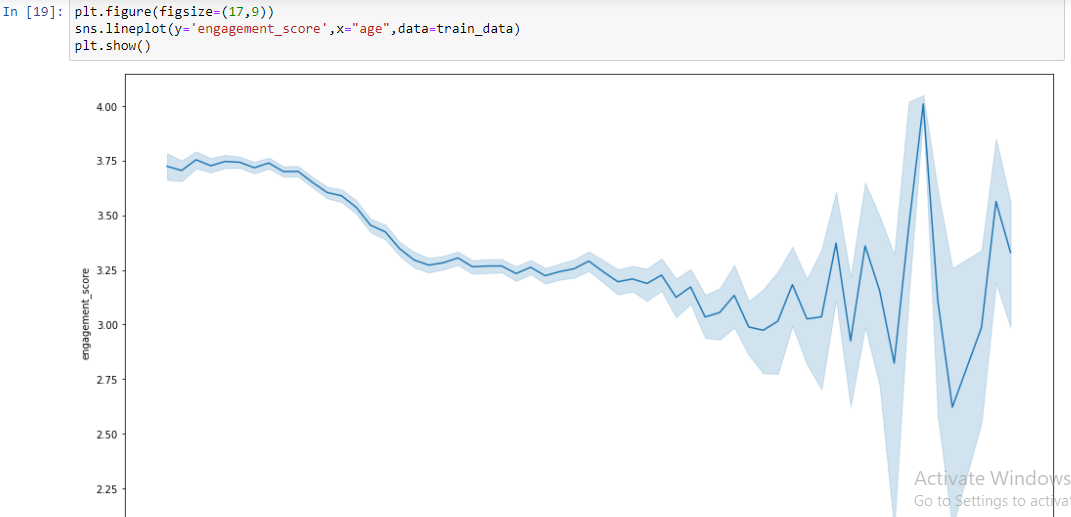












1. **Data Cleaning and feature engineering**

Null values checking :

* There are no null values in training or testing data set

feature engineering:

* I had converted 'gender','profession' into categorical type
* I had dropped 'row\_id','engagement\_score' from training data(X)
* I had dropped target variable 'engagement\_score' from validation data(Y)
* I had splitted the data in 70:30 ratio of train to validation data

1. **ML model selection & development**

Target variable “'engagement\_score'” is **numerical .** So I had to explored all the regression analysis techniques.

I had explored/implemented,

1. Linear Regression
2. Robust Regression
3. Ridge Regression
4. LASSO Regression
5. Elastic Net
6. Polynomial Regression
7. Stochastic Gradient Descent
8. Artificial Neaural Networks
9. Random Forest Regressor
10. **Model Validation**

Among all of above techniques “Random Forest Regressor

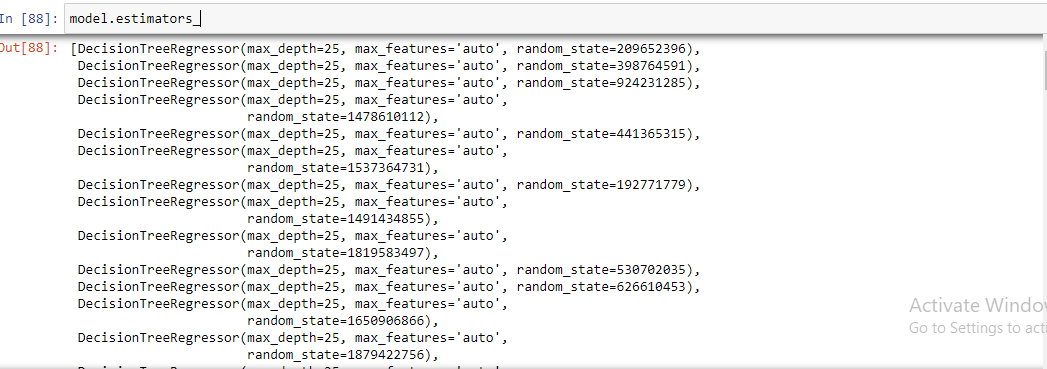
“ was the best because its r2square was hightest .

So I had used that only.

To reach till my final model I had employed 10 different algorithm which I listed in point-3 then again I had tuned “Random Forest Regressor” by (n\_estimators,min\_samples\_split,max\_depth, random\_state)

As well as I had worked on ANN also a lot . I had trained it on 240 epoches(In progressive manner from 10..50..100..140..190..240) but still ANN accuracy was less .

My model structure and how it looks is below:

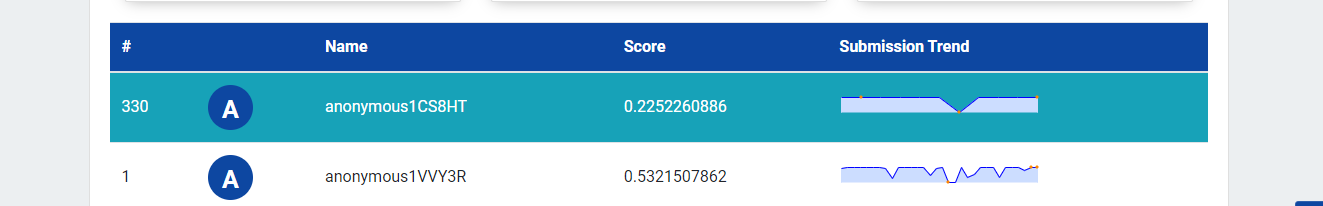


1. **Prediction for test data**

Test data were there in “test\_1zqHu22” file.

I had dropped “row\_id” from it and predicted 'engagement\_score' using different models .

Then I uploaded on Jobathon portal and it gave me **“0.2252260886”** score



1. **Deploy model using flask and docker(Not asked so TBD on demand)**

In future for deployment I can create flask-api and deploy it on docker if any company will want