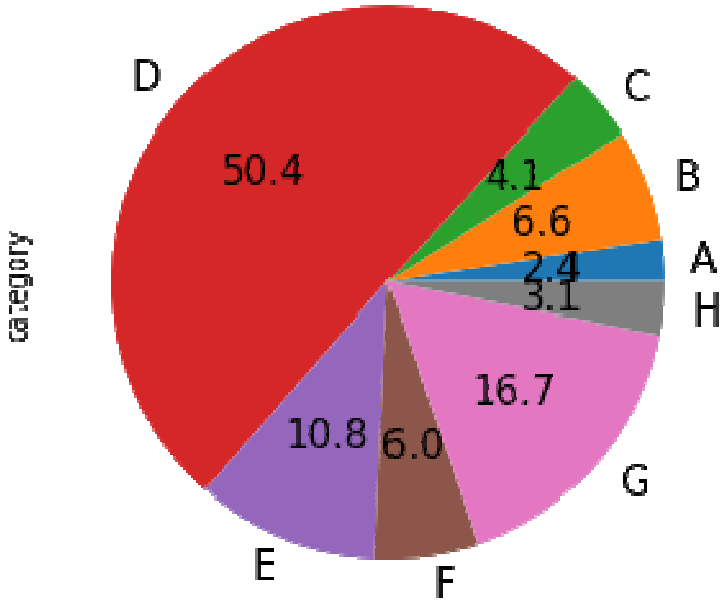
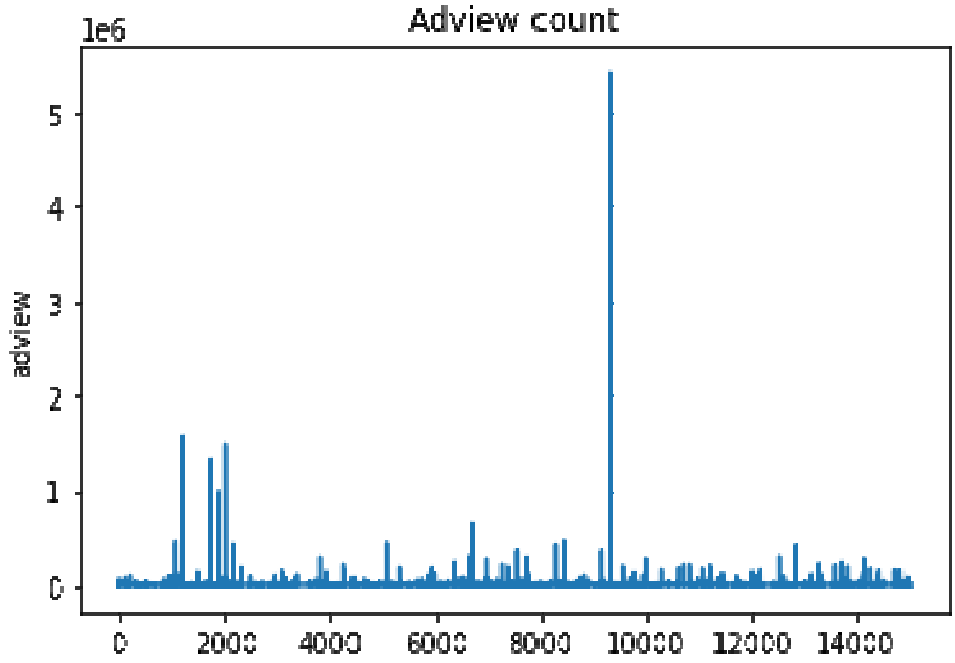
Steps involved during training models:

1. Import the datasets and libraries, and explore the dataset (describe, shape and type of data inside etc):
   - Imported numpy, pandas, matplotlib, seaborn and joblib libraries.
   - Checking the datatype of dataset, datatype of each attribute and null v alues using info.
2. Visualize the dataset using plotting using plots and Heatmap:

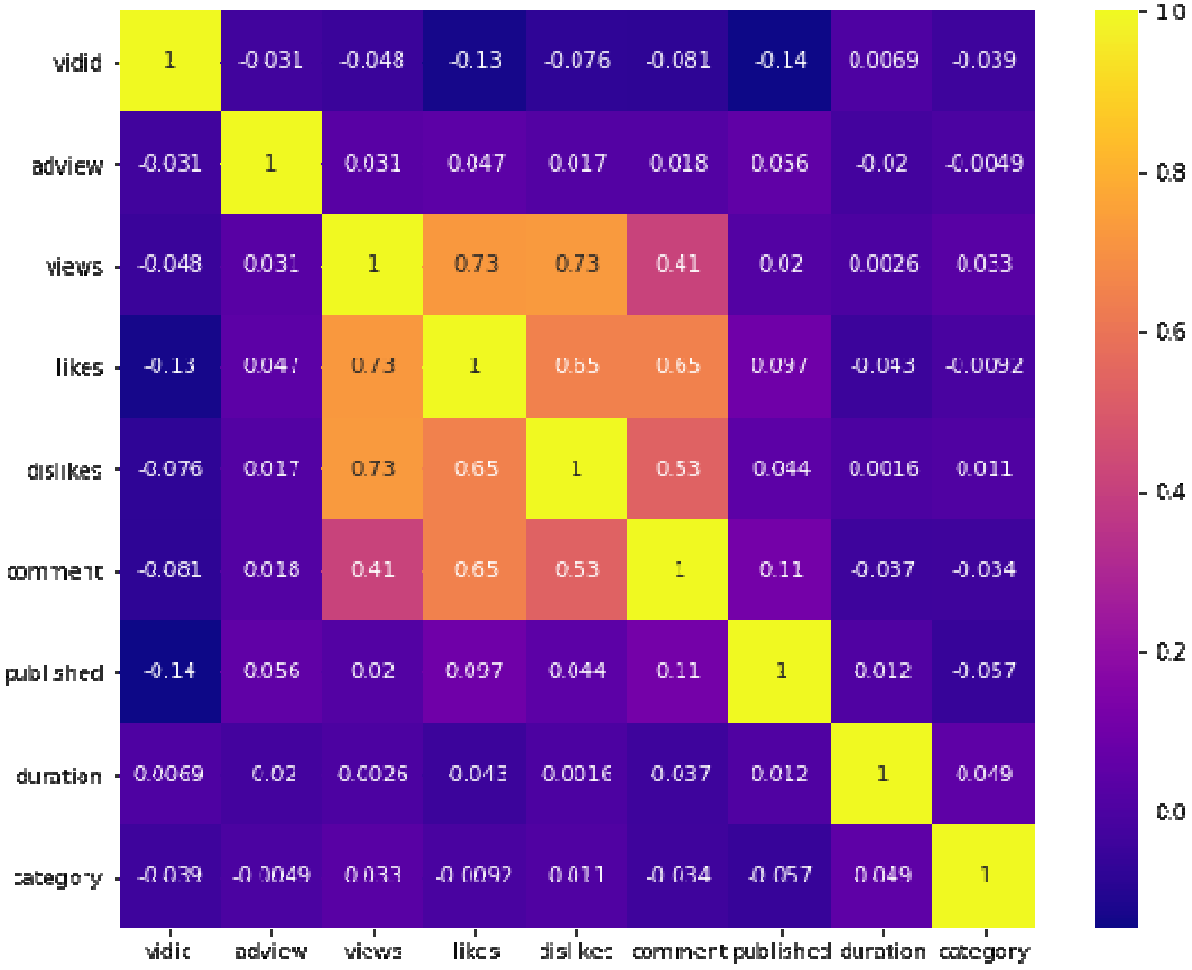
Plots:



The pie chart represent different categories and we find that video of category D are as large as sum of all others



The adview for different dataset is represented above. Here we see that we have some outliner at 8000-10000 so we need to avoid values greater than 2 x 10^6



We see that adview does a visible correlation without other element which we see somewhat for like, dislike, comment and views

3. Clean the data by remove missing values and other unimportant feature or attributes:
   - Remove outliners from adview (greater than 2 x 10^6).
   - Remove character 'F' from views, likes, dislikes and comment.
4. Transform attributes into numerical data (use label encoder for category and date time function for duration and other to numbers using panda library):
   - Convert categories, views, comments, likes, dislikes, adview to numbers using pandas numeric function.
   - Encode duration, vidid and published using labelEncoder.
   - Convert duration to second

5. Normalize your data and split the data into training, validation and test set in the appropriate ratio (say 8:2):
   - Split dataset using test_train_split
   - Normalize both test and train data (used min-max scale).
6. Use following models to train and test data and check errors:
   - Linear regression model
   - Support Vector Regression
   - Random forest model using hyper values
   - Artificial neural network model (having different layers and hyper parameters)
7. Pick the best model based on error as well as generalization

Model Name	Mean Absolute Error	Root Mean Squared Error	Mean Square Error
Linear Regression	3707.38	28907.84	835663131.12
Support Vector ***Regression***	3707.38	28907.84	835663131.12
Decision Tree Regression	2609.10	29739.29	884425468.16
Random Forest Regression	3303.23	25294.80	639827044.59
Artificial Neural Regression	3162.34	28786.91	828686052.52

I have chosen the Decision Tree Regression model as it have least Mean Absolute error than others assuming that big and small error are counted equals. If we want to give big error more penalties, then we can use random forest as it have less root mean squared error.

8. Save your model for future predictions.