Capstone Project-II

TED Talk Views Prediction

By-Prasad Khedkar
(Individual Project)

Points of Discussion :

1. Problem Statement
2. Data Info
3. Data Cleaning
4. Data Processing
5. EDA and Visualization
6. Outlier Treatment
7. Heatmap Co-relation
8. Feature Selection
   & Data Preparation
9. Model Implementation
10. Final Model Selection
11. Hyper-Parameter Tuning

Problem Statement :

TED is devoted to spreading powerful ideas on just about any topic. These datasets contain over 4,000 TED talks including transcripts in many languages. Founded in 1984 by Richard Salman as a nonprofit organization that aimed at bringing experts from the fields of Technology, Entertainment, and Design together, TED Conferences have gone on to become the Mecca of ideas from virtually all walks of life. As of 2015, TED and its sister TEDx chapters have published more than 2000 talks for free consumption by the masses and its speaker list boasts of the likes of Al Gore, Jimmy Wales, Shahrukh Khan, and Bill Gates. **The main objective is to build a predictive model, which could help in predicting the views of the videos uploaded on the TEDx website.**

Data Info :

As can be seen from the image, we have total 19 features or columns.

Among them, 4 are numerical columns- 3 (int64) + 1 (float64) and 15 category columns (object)

And shape of our Dataset i.e., rows x columns is (4005,19)

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	talk_id	4005 non-null	int64
1	title	4005 non-null	object
2	speaker_1	4005 non-null	object
3	all_speakers	4001 non-null	object
4	occupations	3483 non-null	object
5	about_speakers	3502 non-null	object
6	views	4005 non-null	int64
7	recorded_date	4004 non-null	object
8	published_date	4005 non-null	object
9	event	4005 non-null	object
10	native_lang	4005 non-null	object
11	available_lang	4005 non-null	object
12	comments	3350 non-null	float64
13	duration	4005 non-null	int64
14	topics	4005 non-null	object
15	related_talks	4005 non-null	object
16	url	4005 non-null	object
17	description	4005 non-null	object
18	transcript	4005 non-null	object

dtypes: float64(1), int64(3), object(15)

Data Cleaning :

'isnull()' method in pandas is used to get total null values in each column.

And by using dropna method all null values are dropped from each column.

Each row is unique and no duplicates found

```
title              0
speaker_1          0
all_speakers       4
occupations      522
about_speakers   503
views              0
recorded_date      1
published_date     0
event              0
native_lang        0
available_lang     0
comments         655
duration           0
topics             0
related_talks      0
url                0
description        0
transcript         0
dtype: int64
```

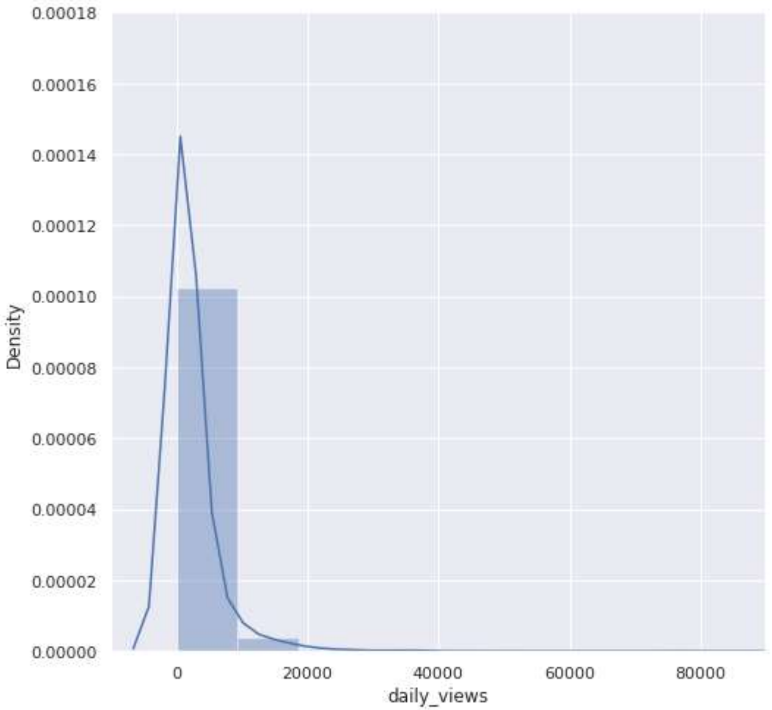
Data Processing :

Added new columns

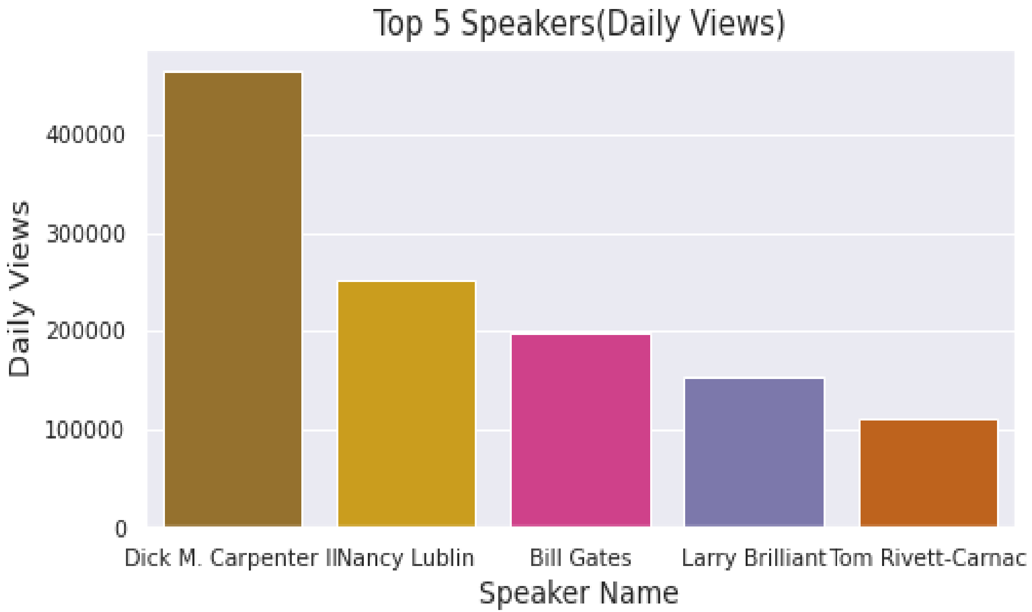
- daily_views
- time_since_published
- speaker_avg_views
- event_avg_views
- number_of_lang_avail
- Total_topics

time_since_published	daily_views	speaker_avg_views	event_avg_views	number_of_lang_avail	Total_topics
5054 days	697	699.75	782.47619	270	9
5054 days	2868	1099.10	782.47619	303	11
5054 days	379	687.75	782.47619	165	9
5054 days	527	453.00	782.47619	219	9

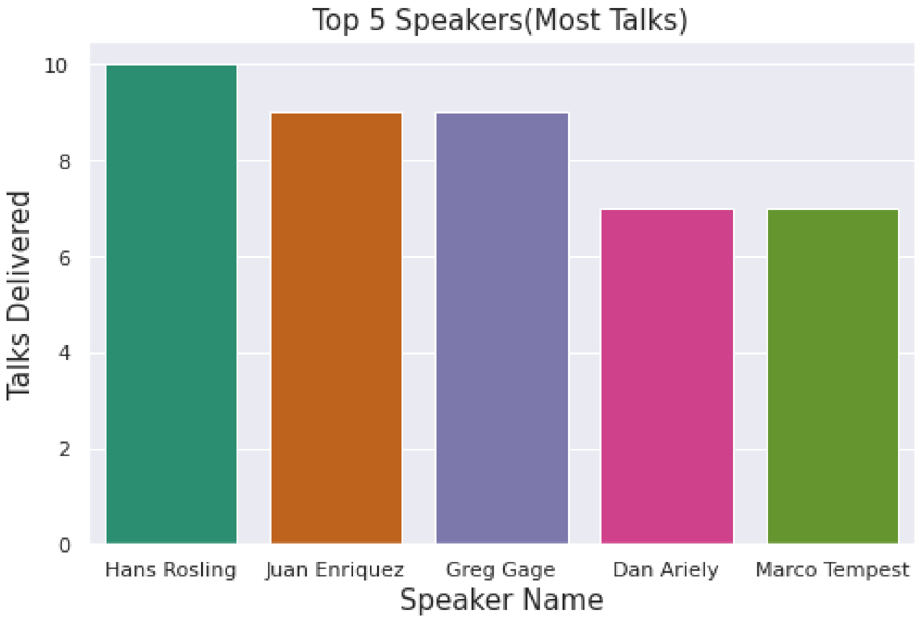
EDA and Visualization :

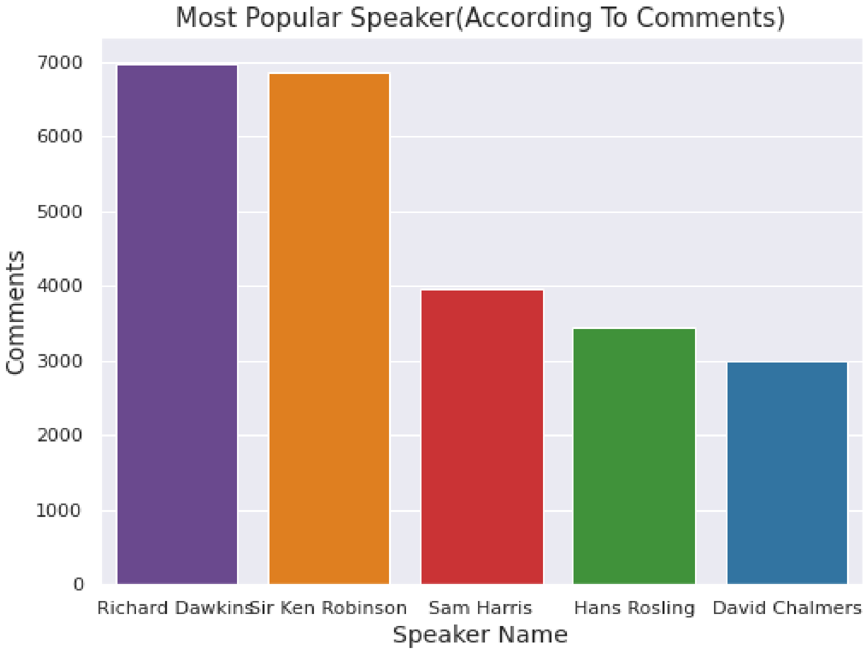


daily_views distplot (positively skewed)

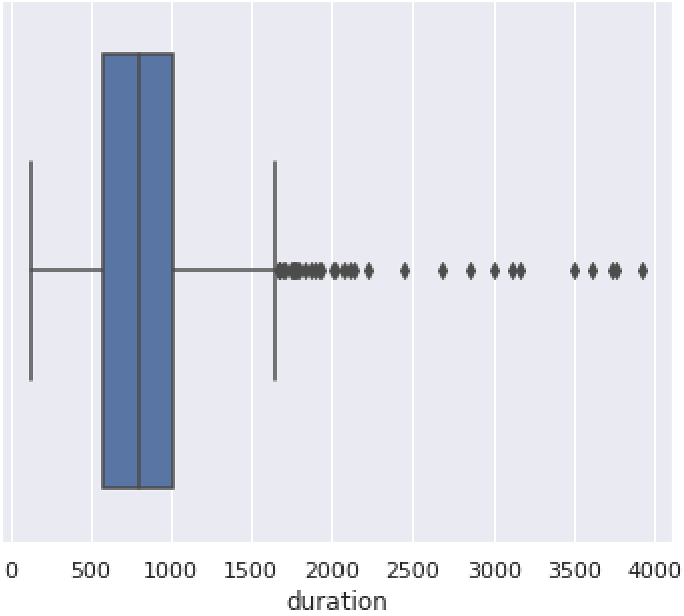


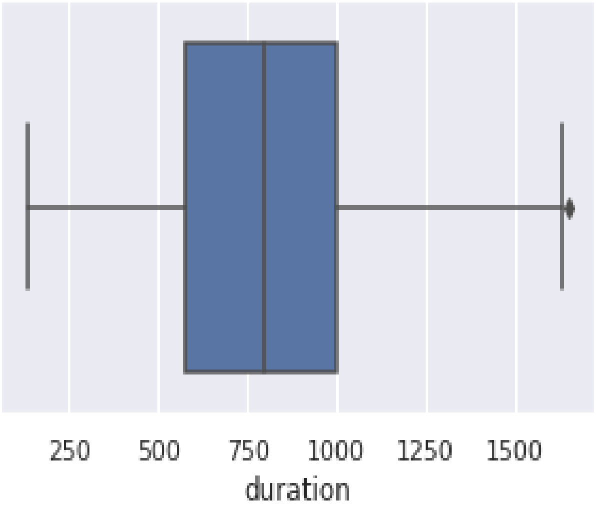
EDA and Visualization :





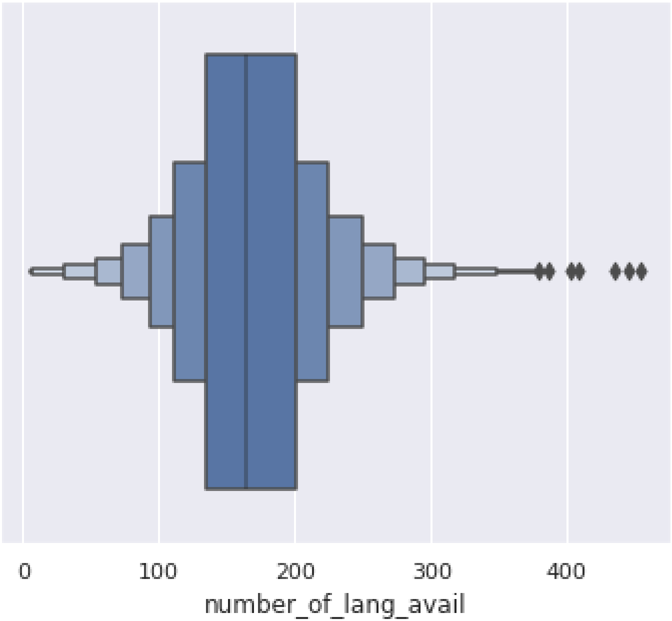
Outlier Treatment :

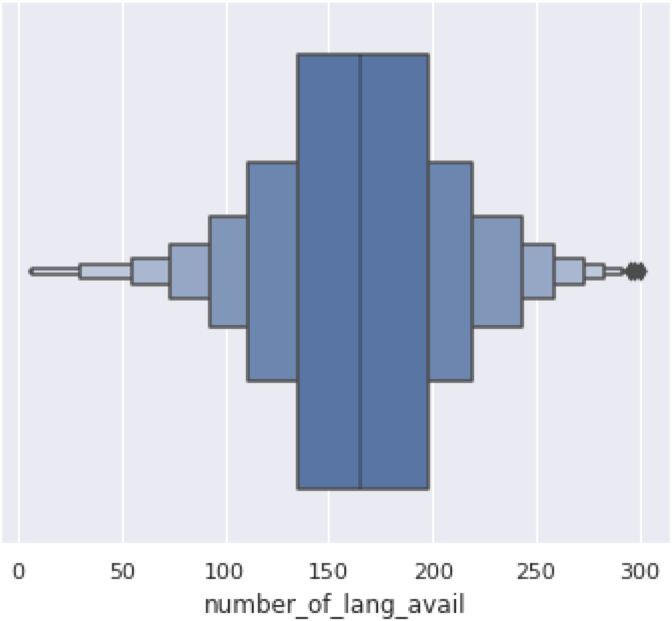




Removing outliers in ‘duration’ feature

Outlier Treatment :





Removing outliers from ‘number_of_lan_avail’ feature

Heatmap Co-relation :

As per the heatmap, 'daily_views' is highly co-related to –

1)'speaker_avg_views'
2)'event_avg_views'

Feature Selection:

Now, important features have to be selected for training our models based on different algorithms.

Independent features :-

duration, speaker_avg_views, event_avg_views, number_of_lang_avail, Total_topics, published_year

Dependent feature :-

daily_views

Data Splitting and Feature Scaling :

Data is splited into - **70% - Training set**

& - **30% - Test set**

StandardScaler is used for scaling the data.

Different Model Implementations :

Different models implemented-

- Simple Linear
- Lasso
- Ridge
- Decision Tree
- SVR
- KNN
- Extra Tree Regressor
- Gradient Boosting
- XGBoost
- Random Forest

Out of these only few performed well-

	Test_R2	Train_R2	Test_RMSE	Train_RMSE	Test_MAE	Train_MAE
Simple Linear	0.80	0.86	-3692.55	-4837.72	-1267.46	-1395.12
Lasso	0.83	0.86	-3412.01	-4855.09	-1201.25	-1350.74
Ridge	0.83	0.86	-3400.79	-4883.25	-1143.85	-1307.40
Random_Forest	0.88	0.89	-1829.71	-4423.59	-501.81	-379.70

Clearly, Random Forest gave top notch scores

Hyper-Parameter Tuning :

After selecting Random Forest as final algorithm to train our model, hyper-parameter tuning (GridSearchCV) is performed to further influence the overall score.

Tuning influenced the Train score from 89% to 93% and Test score from 88% to 89%

	Test_R2	Train_R2	Test_RMSE	Train_RMSE	Test_MAE	Train_MAE
Random_Forest_without_hyper_para	0.88	0.89	-1829.71	-4423.59	-501.81	-379.70
Random_Forest_with_hyper_para	0.89	0.93	-1272.03	-3414.26	-433.15	-342.16

Final Conclusions :

- Random Forest gave the best scores than others and is finalized for model training
- It also gave low RMSE and MAE errors
- With hyper parameter tuning the score increased further
  from 89% to 93% for training dataset &
  from 88% to 89% for test dataset