TED Talk Views Prediction

Documentation

- By Prasad Khedkar
(AlmaBetter Student)

Abstract:

According to Wikipedia, TED is media organisation which posts international talks online under the slogan 'ideas worth spreading'.
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.

Problem Statement:

The main objective is to build a predictive model, which could help in predicting the views of the videos uploaded on the TEDx website.

Dataset Information

- Number of instances: 4,005
- Number of attributes: 19

Features information:

The dataset contains features like:

- **talk_id**: Talk identification number provided by TED
- **title**: Title of the talk
- **speaker_1**: First speaker in TED's speaker list
- **all_speakers**: Speakers in the talk
- **occupations**: Occupations of the speakers
- **about_speakers**: Blurb about each speaker
- **recorded_date**: Date the talk was recorded
- **published_date**: Date the talk was published to TED.com
- **event**: Event or medium in which the talk was given
- **native_lang**: Language the talk was given in
- **available_lang**: All available languages (lang_code) for a talk

- **comments**: Count of comments
- **duration**: Duration in seconds
- **topics**: Related tags or topics for the talk
- **related_talks**: Related talks (key='talk_id',value='title')
- **url**: URL of the talk
- **description**: Description of the talk
- **transcript**: Full transcript of the talk

Introduction :

Ted Talks are one of the institutions that are constantly using Machine Learning algorithms to optimize the number of views the videos receive. They do this by understanding the dependency of views with other relevant features to increase the viewers satisfaction by recommending videos according to the average views that have been affected by features like topics,comments, etc.

Steps involved:

- **Data Collection**

  To proceed with the problem dealing first dataset is loaded that is given, into .csv file into a dataframe then mount the drive and load the csv file into a dataframe.

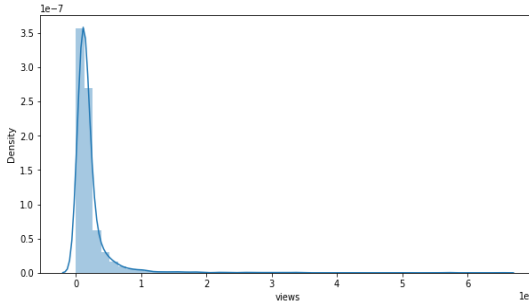
- **Exploratory Data Analysis**
  After loading the dataset duplicate values have to be look after. There were none. So EDA was performed by comparing variable Views with other independent variables. This process helped to figure out various aspects and relationships among the target and the independent variables. It gives a better idea of which feature behaves in which manner compared to the target variable.

  1. **Numerical Variables:**
     - Talk_id
     - Views
     - Comments
     - duration
  2. **Textual Variables:**
     - Title
     - Speaker_1
     - Recorded_date
     - Published_date

- Event
- Native_lang
- Url
- Description

3. **Dictionaries:**
   - Speakers
   - Occupations
   - About_speakers
   - Related_talks
4. **List:**
   - topics

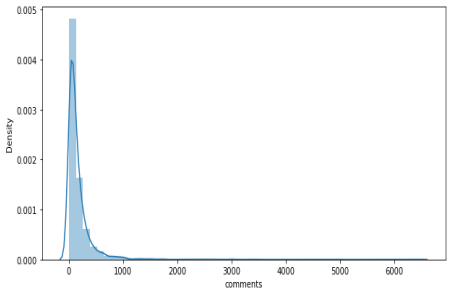
The distplot of views is shown below:



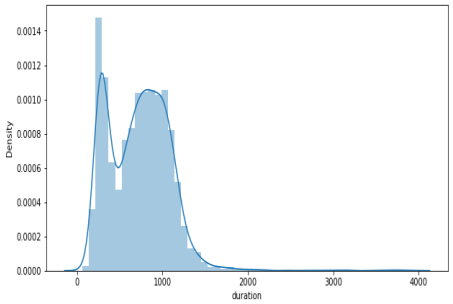
variable ‘views’ was a skewed variable.

The other continuous variables have distributions as:

1. Comments



2. Duration



All of the data had very skewed continuous variable distributions.

- **<u>Null & Duplicate values Treatment:</u>**

The dataset contains 1685 null values which might tend to disturb mean absolute score hence those values are removed by using dropna method in pandas.

There are no duplicate values present in the dataset i.e., every row is unique.

- **Data Processing**

New columns have been added, these are:

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

- **Feature Selection**

In this only important features have been selected those are:

Independent features :-

- *duration*,
- *speaker_avg_views*,
- *event_avg_views*,
- *number_of_lang_avail*,
- *Total_topics*,
- *published_year*

Dependent feature :-

- *daily_views*

- **Outlier Treatment**

Outlier treatment on variables like duration and occupation has been performed. This was done by rremoving outliers with the extreme values at the first and third quartiles.
Outlier treatment is done to prevent high errors that were influenced by outliers.

Fitting different models

Different Models tried are:

1. **Linear Regression**
2. **Lasso Regression**
3. **Ridge Regression**
4. **Decision Tree Regressor**
5. **SVR**
6. **KNN**
7. **Extra Tree Regressor**
8. **Gradient Boosting**
9. **XGBoost Regressor**
10. **Random Forest Regressor**

Out of these only 4 regressors performed well and others overfit on the data

Our shortlisted regressors are

- Linear
- Lasso&Ridge
- Random Forest

Model performance:

The performance of our shortlisted models are as follows:

1] Linear Regression:

- Test R2 score : 0.80
- Train R2 score :0.86
- Test RMSE : -3692.55
- Train RMSE: -4837.72
- Test MAE : -1267.46
- Train MAE:-1395.12

2] Lasso:

- Test R2 score: 0.83
- Train R2 score: 0.86
- Test RMSE: -3412.01
- Train RMSE: -4855.09
- Test MAE: -1201.25
- Train MAE: -1350.74

3] Ridge:

- Test R2 score: 0.83
- Train R2 score: 0.86
- Test RMSE: -3400.79
- Train RMSE: -4883.25
- Test MAE: -1143.85
- Train MAE: -1307.40

4] Random Forest:

- Test R2 score: 0.88
- Train R2 score: 0.89
- Test RMSE: -1829.71
- Train RMSE: -4423.59
- Test MAE: -501.89
- Train MAE: -379.70

As can be seen from above scores, Random Forest performed well on every metrics, be it RMSE or MAE. Also it gave far better Train & Test R2 scores than others.
Therefore, Random Forest is our final algorithm on which final model will be implemented.

Hyper parameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

GridSearchCV is used for hyperparameter tuning. This also results in cross validation and in this case , the dataset is divide into 5- folds.

Grid Search combines a selection of hyperparameters established by

the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations.
The common hyperparameters which were extracted : n_estimators, random_state

Conclusion:

That's it! For this documentation. Starting with loading the data so far I have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.
After hyper parameter tuning, I have prevented overfitting and decreased errors by regularizing and reducing learning rate.
4 Models have performed very well on unseen data due to various factors like feature selection,correct model selection,etc.
Out of those 4, Random Forest performed exceedingly well and is selected for final model implementation.

Future work:

1. We can do a dynamic regression time series modelling due to the availability of the time features.
2. We can improve the views on the less popular topics by inviting more popular speakers.
3. We can use topic modelling to tackle views in each topic separately.

References:

1. AlmaBetter videos & notes
2. Towards data science website
3. GeeksforGeeks
4. Analytics Vidhya