Quantitative prediction of popularity of ted topics among TEDster

| Gulwish Dilawar |
| --- |
| *Department of Software Engineering* |
| *University of Pakistan* |
| 19sw39@students.muet.edu,pk |

1. ***Abstract* - TED Talks are a great source of freely available knowledge and ideas online. TED Talks cover many topics presented by a variety of speakers, including technology, entertainment, design, culture, and academic research. The purpose of this study is to develop a model that predicts the number of views of the  TED talk .We used several machine learning Regression  algorithms such as random forest , ExtraTreesRegressor ,XGboost regressor  to propose the most accurate regression algorithm for  selected predictors. Dataset contains details of 2550 TED talks from 2006 to 2017.**

***Index Terms – regression ,Ted talk , predictors ,EDA***

I. Introduction

The TED talks vary in length, from 3 to 30 minutes, but are assigned single word labels by raters, representing the overall impression of the talk. This  qualitative prediction of  the  talk impact  is not efficient enough in gaining the information about demanding topics  since it is not efficient enough instrategizing campaigns for events, brand marketing **Because** Information about what are the most popular topics serves as indicators  of public's thoughts and viewpoint which is highly demanded due to its various applications.

In our study, we aim to identify such popular topics of  TED talk ,which is a platform  aim at spreading powerful ideas on just about any topic, based on view counts because the  the estimation of quantitative talk impact is efficient in understanding TEDster viewpoint and concerns Duration ,delivered talks , comments ,event type

**Approach taken :**

The task was divided into 2 main parts :

1. Statistical Analysis over the dataset to discover relationships between each feature and the target variable . So that this relationship information can be used by the management in making better Business decisions
2. Creating a Machine Learning Pipeline , that can take in the data of any new video and predict how many views it will generate on a daily basis .It was required to keep this pipeline modular , such that it can be retrained often when new data is collected.

The paper proceeds as follows. The literature review section focuses on having an overview of the research background. The Methodology The method section introduces our rationale for using regression models. The result section displays different models’ performance and the learned parameter importance. Finally, the conclusion section summarizes all the work we have done in this research and looks into possible exploration in the future.

II. Literature Review

predicted humorous utterances using audience’s laughter based on TED talks[1]; focused on predicting the popularity of micro-videos on Vine using four indicators: the number of comments, the number of likes, the number of reposts and the number of views [2]; targeted on Twitter and measured a tweet’s popularity by the number of its future retweets[3]; [4] defined an image’s popularity by its view count and the number of comments, etc. Besides, not surprisingly, on top of common popularity prediction for videos, social media, and images, almost every kind of web content’s popularity has been measured for exploration, such as online news [5], even for Github repositories [6]. To sum up, based on prior experience, we have learned that almost every type of web content’s popularity can be measured by the frequency of certain kinds of human interaction with it, which in the TED talk’s case could be the number of views or the number of comments. At the same time, we take the “accumulating effect” into account, given a longer time of an item remaining online naturally triggers more human interactions with it. Therefore, we think it would be more reasonable to use the number of averaged views or comments of each TED talk in a certain period as the indicator of their popularity. Possible Predictors for “Popularity”.TED talks in a certain period as an indicator of their popularity. Possible Predictors for “Popularity”.

II. Methodology

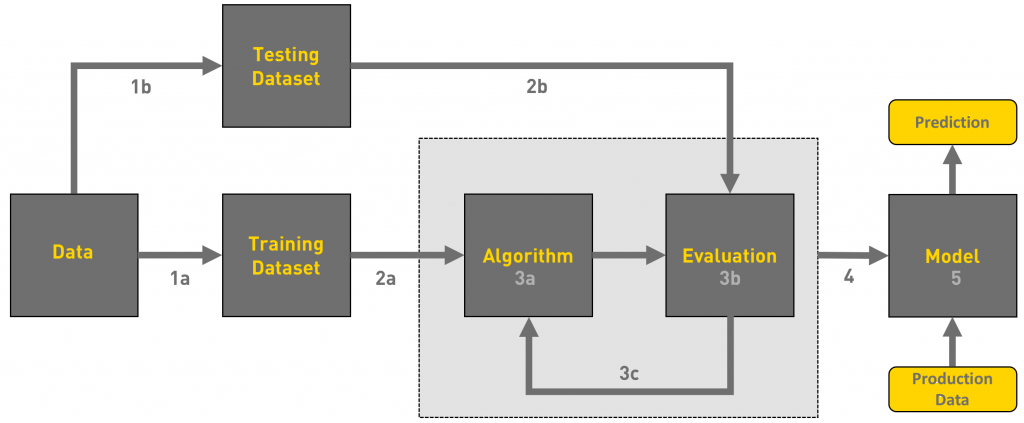


Fig 1: Working flow of model building

1. ***Dataset details***

The original dataset on Kaggle contains all 2,550 TED talks published on the official TED.com website from February 24th, 2006 to September 22nd, 2017.

**Dataset info**

* Number of records: 4,005
* Number of attributes: 19

To perform well on predicting TED Talks video views,

| ['comments', 'description', 'duration', 'event', 'film\_date', 'languages', 'main\_speaker', 'name', 'num\_speaker', 'published\_date', 'ratings', 'related\_talks', 'speaker\_occupation', 'tags', 'title', 'url', 'views'] |
| --- |

- **\*\*duration\*\*** - duration of the video

- **\*\*event\*\*** - name of the event of which the talk is part of

- **\*\*languages\*\*** - number of languages in which the talk is available in

- **\*\*num\_speaker\*\*** - number of speakers in the talk

- **\*\*film\_date\*\***, **\*\*published\_date\*\*** - date of filming and publishing the talk, from which I extract:

- **\*\*day of the week\*\***

- **\*\*month\*\***

- **\*\*year\*\***

- **\*\*related-talks\*\*** - an array that consists of 6 related talks, from which I extract the average number of views

The point of the task is to predict the number of views for a video which has just been released or is yet to be released. After going through the data analysis notebooks I mentined earlier, I decided to exclude the following features:

- **\*\*comments\*\*** - number of comments on the video

- **\*\*ratings\*\*** - number of times the video has been rated

- **\*\*name\*\*** - name of the talk, which includes the name main speaker and title of the talk

- **\*\*main speaker\*\*** - name of the main speaker that leads the talk, we rarely see the same speaker do more than 1 talk

- **\*\*title\*\*** - title of the talk

- **\*\*url\*\*** - url link to the talk

The following features I leave for future work:

- **\*\*description\*\*** - description of the talk, will need to encode this information

- **\*\*tags\*\*** - tags that are associated with the talk

- **\*\*speaker\_occupation\*\*** - occupation of the main speaker

**Target Variable :**

* **views**: Contains Count of views of every talk

1. *Training Dataset*

**One-Hot-Encoding** on the categorical attributes and will have the data ready for training machine learning models using get dummies function

1. *Testing* Datasets

X\_train, X\_test, y\_train, y\_test**=** train\_test\_split(X, y, test\_size**=** 0.20, random\_state**=** 5)

once we have decided on a certain model, we need the test dataset to report how the selected model can generally perform on the data outside our model building.

Specifically speaking, we will use training and validation datasets to train our models and select the best one from them based on their different levels of prediction performances, which can be reflected by mean squared error (MSE).

1. ***Algorithm and Evaluation and Model Prediction***

Dataset is split in training (80%) and test (20%) sets.

Used **Mean Absolute Error (MAE)** to measure the error as it will more intuitive understanding of how accurate the model is and *hyper parameter tuning is done using randomized search cv*

n\_estimators **=** [50,80,100]

*# Maximum depth of trees*

max\_depth **=** [4,6,8]

*# Minimum number of samples required to split a node*

min\_samples\_split **=** [50,100,150]

*# Minimum number of samples required at each leaf node*

min\_samples\_leaf **=** [40,50]

*# HYperparameter Dict*

param\_dict **=** {'n\_estimators' : n\_estimators,

'max\_depth' : max\_depth,

'min\_samples\_split' : min\_samples\_split,

'min\_samples\_leaf' : min\_samples\_leaf}

1. **Random Forest** is used as baseline model to start testing

OUTPUT :

Training MAE: 458747.60

Test MAE: 646002.69

Views mean: 1698297.48

Views std: 2498479.37

Plotting the feature importance that are derived from the model that was just trained

.

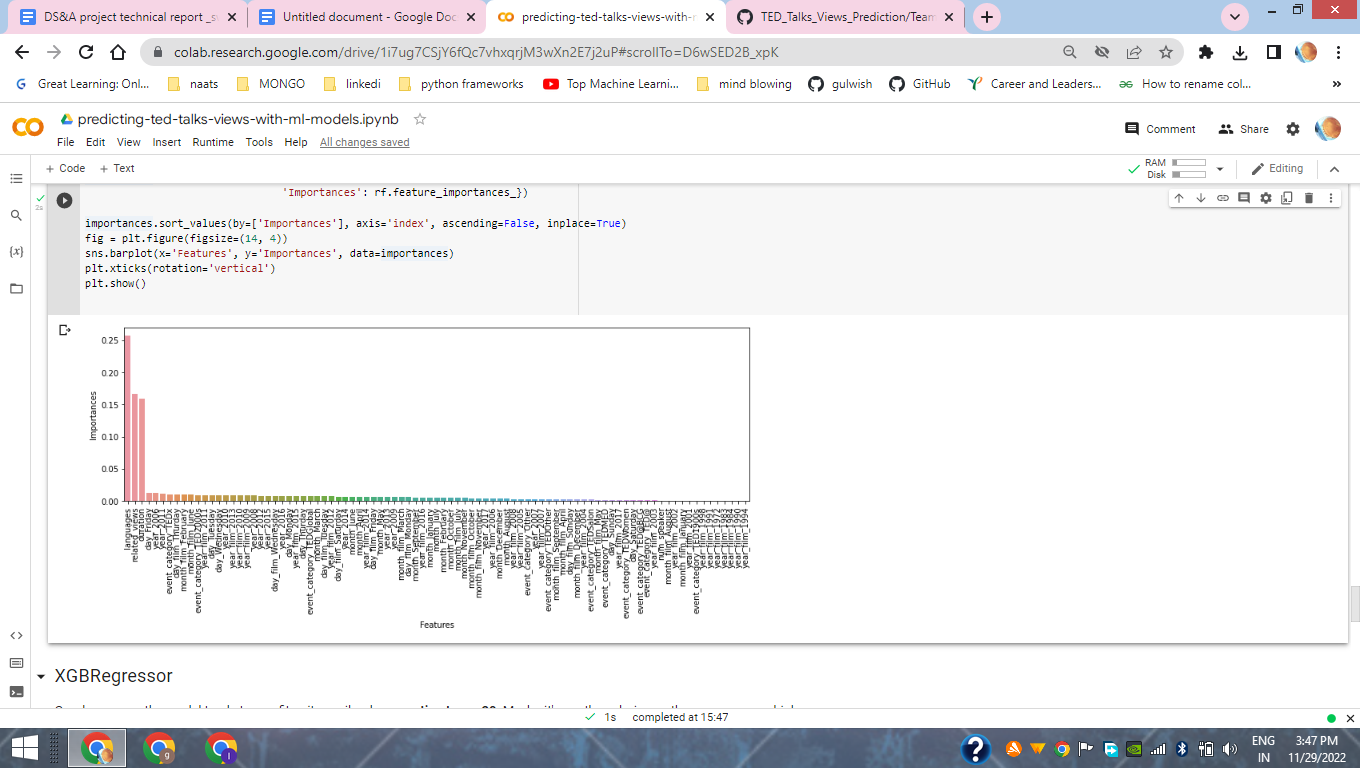


Fig 4 : predictors

1. **XGboost Regressor** the model tends to overfit quite easily when **n\_estimators > 20**. Maybe it's worth exploring wether we can use a higher **n\_estimators** value while using the other hyper parameters to regularize the model.

| xgbr = xgb.XGBRegressor(criterion='mae', earning\_rate=0.1, max\_depth=10, subsample=0.5, n\_estimators=20, min\_child\_weight=2, random\_state=2019)  xgbr.fit(X\_train, y\_train)  y\_pred = xgbr.predict(X\_train)  y\_test\_pred = xgbr.predict(X\_test)  print('Training MAE: {:0.2f}'.format(metrics.mean\_absolute\_error(y\_train, y\_pred)))  print('Test MAE: {:0.2f}'.format(metrics.mean\_absolute\_error(y\_test, y\_test\_pred))) |
| --- |

OUTPUT

Training MAE: 495106.41

Test MAE: 640284.68

1. **ExtraTreesRegressor**

since the gap between Training and Test MAE is quite big to regularize the model this model is used

OUTPUT

Training MAE: 265596.88

Test MAE: 594999.50

1. ***Results and Performance Evaluation***

Figure The ML repressor models that have been used are

| **Name** | **MAE\_train** | **MAE\_test** |
| --- | --- | --- |
| RandomForest | 186583.315347 | 191844.536467 |
| ExtraTreeRegressor : | 207304.048833 | 204793.751052 |
| XGBRegressor: | 164091.332037 | 226944.860549 |

1. ***Final selection of the model***

We choose MAE and not RMSE as the deciding factor of our model selection because of the following reasons:

* RMSE is heavily influenced by outliers as in the higher the values get the more the RMSE increases.
* MAE doesn’t increase with outliers. MAE is linear and RMSE is quadratically increasing.
* The best performing regressor model for this dataset is Random Forest Regressor on the basis of MAE**.**

1. *CONCLUSION*

Number citations consecutively in square brackets [1]. Punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]. Use “Ref. [3]” or “Reference [3]” at the beginning of a sentence: “Reference [3] was the first …”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes (see Table I). *IEEE Transactions* no longer use a journal prefix before the volume number.

Started with loading the data so far we have done EDA ,feature engineering , data cleaning, target encoding and one hot encoding of categorical columns, feature selection and then model building.

Following models were used

* Random Forest Regressor
* Extra Tree Regressor
* XGB Regressor

In all of these models our errors have been in the range of 2,00,000 which is around 10% of the average views. We have been able to correctly predict views 90% of the time.

After hyper parameter tuning, overfitting is prevented due to and decreasing errors by regularizing and reducing learning rate.

Given that only have 10% errors, all models have performed very well on unseen data due to various factors like feature selection,correct model selection,etc.

Out of all these models RandomForestRegressor is the best performer in terms of MAE.

In all the features speaker\_wise\_avg\_views is most important this implies that speakers are directly impacting the views**.**

IV. **Future work**

We can do a dynamic regression time series modeling due to the availability of the time features.

We can use topic modeling to tackle views in each topic separately**.**

References

1. M. King, B. Zhu, and S. Tang, “Optimal path planning,” *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
2. H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
3. M. King and B. Zhu, “Gaming strategies,” in Path Planning to the West, vol. II, S. Tang and M. King, Eds. Xian: Jiaoda Press, 1998, pp. 158-176.
4. B. Simpson, et al, “Title of paper goes here if known,” unpublished.
5. J.-G. Lu, “Title of paper with only the first word capitalized,” *J. Name Stand. Abbrev.*, in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” *IEEE Translated J. Magn. Japan*, vol. 2, pp. 740-741, August 1987 [*Digest 9th Annual Conf. Magnetics Japan*, p. 301, 1982].
7. M. Young, *The Technical Writer’s Handbook*, Mill Valley, CA: University Science, 1989.