**TED Talk Views Prediction**

**Anjali Tidke, Shubham Dukare,**

**Data science trainees,**

**AlmaBetter, Bangalore**

**Abstract:**

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.

***Keywords: machine learning, views prediction, TED Talk, Regression model***

**1.Problem Statement**

The main goal of the project is to build a predictive model, that could help us to predict the views of the videos uploaded on TEDx Website.

**Column (Feature) information:**

1. **talk\_id**: Talk id number provided by TED
2. **title**: Title of the talk
3. **speaker\_1**: First speaker in TED's speaker list
4. **all\_speakers**: Speakers in the talk
5. **occupations**: Occupations of the speakers
6. **about\_speakers**: Blurb about each speaker
7. **recorded\_date**: Date the talk was recorded
8. **published\_date**: Date the talk was published to TED.com
9. **event**: Event or medium in which the talk was given
10. **native\_lang**: Language of the talk
11. **available\_lang**: All available languages for a talk
12. **comments**: Comments
13. **duration**: Duration in seconds
14. **topics**: Topics for the talk
15. **related\_talks**: Related talks
16. **url**: URL of the talk
17. **description**: Description of the talk
18. **transcript**: Transcript of the talk
19. **views**: The number of views for each video. This will be our target variable.

**2. Introduction**

### TED Talks are one of the most influential videos on the internet, where experts speak on education, business, science, tech, and creativity. We have data of this TED Talks videos. We have to analyze the data, explore the data, find out the features, apply regression models.

### We have to find some features from the data by which TED Talks can increase the views on their videos.

**3. Project Workflow:**

1. Importing Libraries
2. Loading the Dataset
3. EDA on features
4. Feature Engineering
5. Data Cleaning
6. Feature selection
7. Fitting the regression models and HyperParameter Tuning
8. Comparison of Models
9. Final selection of the model
10. Conclusion

**4. Steps involved:**

* **Importing libraries and data loading**

We have to import the required libraires and load the data into the python dataframe object.

The data provided is in the form of csv file**.**

* **Perform Exploratory Data Analysis**

After loading the dataset, We have to check the size of the dataset. The dataset has 4005 entries with 19 columns.

By performing dataset.info and dataset.head operation. We get some inputs about the column names. We can see, views will be our target variable. As we have to perform the analysis, so data type of every column is important factor to check.

1. **Numerical Variables:**

* Talk\_id
* Views
* Comments
* Duration

1. **Textual Variables:**

* Title
* Speaker\_1
* Recorded\_date
* Published\_date
* Event
* Native\_lang
* Url
* Description

1. **Dictionaries:**

* Speakers
* Occupations
* About\_speakers
* Related\_talks

1. **List:**

* topics

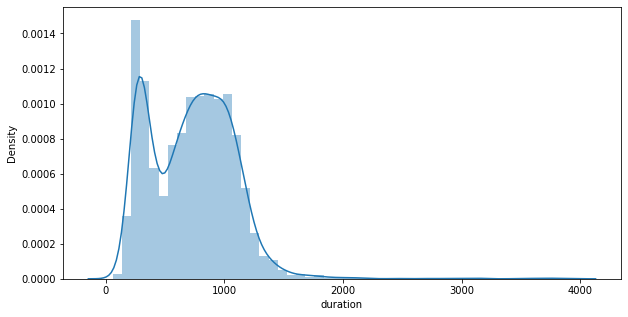
talk\_id and url column will not be important for analysis, it will be better if we drop it.

As we can see above, The target variable ‘views’ was a skewed variable.

1. Comments



1. Duration



All of the data had very skewed continuous variable distributions.

* **Handling Null values**

Occupations, about\_speakers, comments and recorded\_date column contains the null values.

We chose to impute nan values and not drop them due to the size of the data set.

* **Encoding of categorical columns**

We had used the Target Encoding for replacing the values of categorical variables with the mean of the views. This was performed to not increase the dimensions to the data set while also keeping the relationship of variables with views into consideration.

* **Feature Selection**

To perform the feature selection, We have combind some features into new features and also created some new features. Also, we have used f\_regression in which we have taken the features with the maximum f-scores.

* **Outlier Treatment**

We had performed outlier treatment to remove the high errors which will be generated by outliers. We have done outlier treatment by replacing the outliers with extreme values.

* **Fitting different models**

For modelling we tried various regression algorithms like:

1. **Random Forest Regressor**
2. **XGBoost Regressor**
3. **Extra Trees Regressor**

* **Hyperparameter Tuning**

Hyperparameter Tuning is necessary to generate less error values and to avoid overfitting for the tree-based models.

**4.1. Algorithms:**

We have used only non-parametric models for prediction because two of the hypotheses such as linearity between output and input variables and errors normally distributed were not met.

1. **Random Forest Regressor:**

Bootstrapping is the process of randomly sampling subsets of a dataset over a given number of iterations and a given number of variables. These results are then averaged together to obtain a more powerful result. Bootstrapping is an example of an applied ensemble model.

The bootstrapping Random Forest algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a new result that often leads to strong predictions/classifications.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

Some of the important parameters while using Random Forest Regressor model are given below:

**n\_estimators** — the number of decision trees you will be running in the model

**criterion** — this variable allows you to select the criterion (loss function) used to determine model outcomes. We can select from loss functions such as mean squared error (MSE) and mean absolute error (MAE). The default value is MSE.

**max\_depth** — this sets the maximum possible depth of each tree

**max\_features** — the maximum number of features the model will consider when determining a split

**bootstrap** — the default value for this is True, meaning the model follows bootstrapping principles (defined earlier)

**max\_samples** — This parameter assumes bootstrapping is set to True, if not, this parameter doesn’t apply. In the case of True, this value sets the largest size of each sample for each tree.

The result of this regressor given below:

MAE train: 186605.126209

MAE test: 191910.559875

R2\_Score train:0.806218

R2\_Score test: 0.803192

RMSE\_Score\_train:485340.428182

RMSE\_Score\_test:488994.200362

1. **XGBoost Regression:**

XGBoost is an efficient implementation of gradient boosting that can be used for regression predictive modeling.

Extreme Gradient Boosting, or XGBoost for short, is an efficient open-source implementation of the gradient boosting algorithm. It is computationally effectively faster with better model performance.

**n\_estimators**: The number of trees in the ensemble, often increased until no further improvements are seen.

**max\_depth**: The maximum depth of each tree, often values are between 1 and 10.

**eta**: The learning rate used to weight each model, often set to small values such as 0.3, 0.1, 0.01, or smaller.

**subsample**: The number of samples (rows) used in each tree, set to a value between 0 and 1, often 1.0 to use all samples.

**colsample\_bytree**: Number of features (columns) used in each tree, set to a value between 0 and 1, often 1.0 to use all features.

XGboost can be optimized by fixing the number of trees, fixing learning rate, tuning gamma, tuning regularization and various hyper parameter tuning.

The result after implementing this model is shown below:

MAE train: 163241.112787

MAE test: 224763.569788

R2\_Score train: 0.918432

R2\_Score test: 0.832182

RMSE\_Score train: 314883.354708

RMSE\_Score test: 451546.48760

.

1. **Extra Trees Regressor:**

Extra Trees is an ensemble machine learning algorithm that combines the predictions from many decision trees.

It is related to the widely used random forest algorithm. It can often achieve as-good or better performance than the random forest algorithm, although it uses a simpler algorithm to construct the decision trees used as members of the ensemble.

The random selection of split points makes the decision trees in the ensemble less correlated, although this increases the variance of the algorithm. This increase in variance can be countered by increasing the number of trees used in the ensemble.

We use the criterion as ‘MAE’ as it uses L1 regularization to select the median and selects the best features for reducing the mean absolute error.

MAE is used as it is not influenced by outliers.

The results obtained after implementing this model are:

MAE\_train : 207236.230507

MAE\_test: 204673.390587

R2\_Score\_train: 0.796772

R2\_Score\_test:0.806601

RMSE\_Score\_train:497028.743503

RMSE\_Score\_test:484741.159212

**4.2. Model performance:**

Model can be evaluated by various metrics such as:

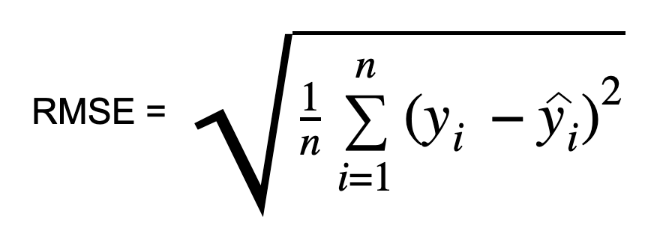
1. **Root Mean Square Error**-

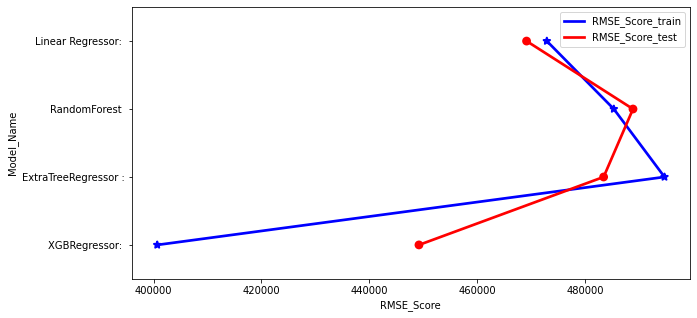
RMSE is computed by taking the square root of MSE. RMSE is also called the Root Mean Square Deviation. It measures the average magnitude of the errors and is concerned with the deviations from the actual value. RMSE value with zero indicates that the model has a perfect fit. The lower the RMSE, the better the model and its predictions. A higher RMSE indicates that there is a large deviation from the residual to the ground truth.

One major drawback of RMSE is its sensitivity to outliers and the outliers have to be removed for it to function properly.

RMSE increases with an increase in the size of the test sample. This is an issue when we calculate the results on different test samples.

Like MSE, RMSE is dependent on the scale of the data. It increases in magnitude if the scale of the error increases.

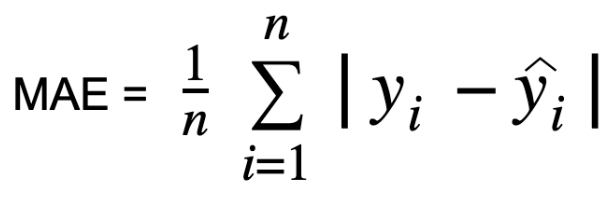




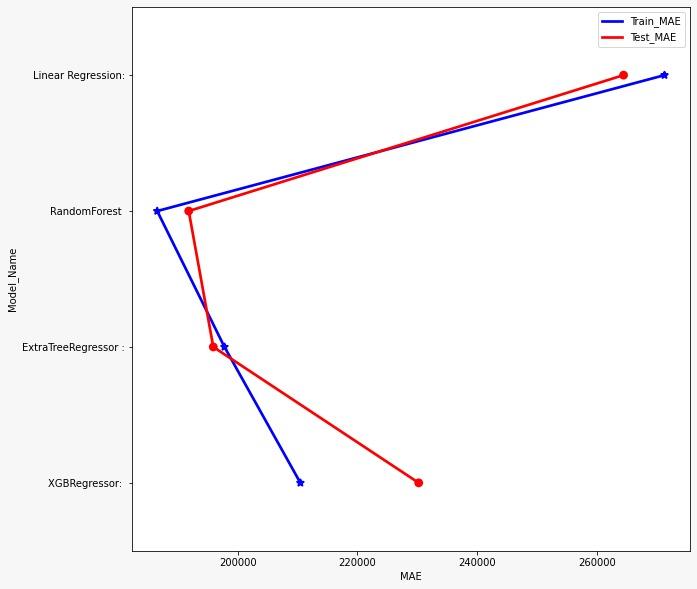
1. **Mean Absolute Error**-

Mean absolute error, also known as L1 loss is one of the simplest loss functions and an easy-to-understand evaluation metric. It is calculated by taking the absolute difference between the predicted values and the actual values and averaging it across the dataset. Mathematically speaking, it is the arithmetic average of absolute errors. MAE measures only the magnitude of the errors and doesn’t concern itself with their direction. The lower the MAE, the higher the accuracy of a model.

Mathematically, MAE can be expressed as follows:



Where, y\_i = actual value, y^\_i = predicted value, n = sample size



As we know, RMSE is more influenced by outliers MAE doesn't increase with outliers.

MAE is linear and RMSE is quadratically increasing.

So, We chosen MAE as a deciding factor for our model.

On the basis of MAE,

The best performing regression model is **Random Forest Regressor.**

After hyper parameter tuning, we have prevented overfitting and decreased errors by regularizing and reducing learning rate.

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

**5. Conclusion**

That's it! we find out which model is suitable.

we have Started with data loading and we have done EDA, feature engineering, data cleaning, target encoding feature selection and then model building.

So far we have used this models:

Ridge Regressor

Lasso Regressor

KNearestNeighbors Regressor

Gradient Boosting Regressor

Random Forest Regressor

Extra Tree Regressor

XGB Regressor

As we know, RMSE is more influenced by outliers MAE doesn't increase with outliers.

MAE is linear and RMSE is quadratically increasing.So, We chosed MAE as a deciding factor for our model.

On the basis of MAE, The best performing regression model is Random Forest Regressor.

we have Successfully build a predictive model, which is helps to TED in predicting the views of the talks uploaded on the TEDx website

TED can increase their views and popularity by increasing videos on sections like Science and Technology.

TED can use topic modelling to tackle views in each topic separately.

**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. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya