

Ted-talk Views Prediction

Ankit Kumar, Fahad Mehfooz,

Sanjog Mishra, Varun Nayyar

Data science trainees

AI maBetter, Bangalore

Abstract:

This paper developed a model that predicts the number of views in a TED-talk video. TED started 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.

With the prediction of the number of views, we can also predict the popularity of a particular speaker.

This study and model implementation is done using the data provided. The inferences from the data is observed through Exploratory Data Analysis. Most of the time is utilised in Data preparation including wrangling, preprocessing, feature engineering. After that the data is fitted into different Machine Learning Algorithms. The metric scores from the different models are compared with each other. To improve the accuracy more, models are optimized with Hyperparameter tuning. Finally, the best fit model is considered to predict the number of views in a video.

Keywords: Supervised Learning, Regression, Elastic net, Random Forest, XGboost, LGBM, Catboost, MSE, RMSE, MAE, R-squared, Adjusted R-squared

1. Introduction

TED (Technology, Entertainment, Design) is a media organization which posts talks online for free distribution, under the slogan “*ideas worth spreading*”. TED was conceived by [Richard Saul Wurman](#), who co-founded it with [Harry Marks](#) in February 1984 as a conference; it has been held annually since 1990. TED's early emphasis was on technology and design. It has since broadened its perspective to include talks on many scientific, cultural, political, humanitarian and academic topics. Since June 2006, TED Talks have been offered for free viewing online, and till now these videos have been watched more than a billion times. Currently, TED is running different events and projects which includes TEDGlobal, TEDx (independent events similar to TED and can be organized by anyone who obtains a free license from TED, and agrees to follow certain principles), TED-Ed (YouTube channel of Ted which creates short animated educational videos), TEDMED, TEDYouth, TEDWomen, TEDSalon.

In this project, the TED dataset has been provided that consists of text and numeric data containing hundreds of tuples.

Mining the information out of this dataset will provide us with some interesting insights about the behavior of the viewers and about presenter's selections about a particular topic. Gaining knowledge about these key aspects will guide us in exploring some interesting facts, relationships, and patterns in different attributes.

2. Problem Statement:

TED is devoted to spreading powerful ideas on just about any topic. The dataset contains over 4,000 TED talks including transcripts in many languages. The main objective is to build a predictive model, which could help in predicting the views of the videos uploaded on the TED website.

3. Data Features

- **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 delivering the talk
- **occupations**: Occupations of the speakers
- **about_speakers**: Information about each speaker
- **recorded_date**: Date on which the talk was recorded
- **published_date**: Date on which the talk was published to TED.com

- **event**: Event or medium in which the talk was given which are different TED projects/events.
- **native_lang**: Language in which the talk was given in
- **available_lang**: Languages (lang_code) in which the talk is available.
- **comments**: Count of comments in each video.
- **duration**: Duration in seconds
- **topics**: Related tags or topics discussed in the talk
- **related_talks**: Related talks means talks which are similar to the video. (key='talk_id',value='title')
- **url**: URL of the talk
- **description**: Description of the talk
- **transcript**: Full transcript of the talk
- **views**: Count of views in each video.

4. Steps involved:

● Assessing Data

Assessing all the columns and trying to get some insights from it. Then there should be checking of null values and duplication of data. Since the dataset we are provided contain null values in some of the columns which needs to be resolved. The features are divided in following groups:

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

● Exploratory Data Analysis

After assessing, the data column is analyzed individually through Univariate analysis. The results obtained from that become the base of our further experimentation which is done through bivariate and multivariate analysis. The target response 'views' is compared with other independent variables

This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

● Null value treatment

Our dataset contains some null values which might tend to disturb

our mean absolute score hence we have performed KNN nan value imputer for numerical features and replaced categorical features nan values with the value 'Other'. We chose to impute nan values and not drop them due to the size of the data set.

● Train-Test-Split

Before doing any feature engineering like encoding and scaling, the data must be split into a train and test set. All those feature engineering must be done on a train set. The test data must be unseen in order to predict accurately.

● Data preprocessing and Feature engineering

This step includes preprocessing of data before fitting into the machine learning model. Some features which are not really important for the prediction like talk_id, url, so they were excluded. Those columns which are in the form of date are segregated into day, month and year to generate some insights. Some feature extraction is also done by adding some new features by combining existing features. Features which are highly skewed are standardized using StandardScaler preprocessing features. Features which are in the textual format are converted in the form vector array using word2vector embedded

feature. Word2vector uses google's pre-trained word2vec model. It is applied on the transcript feature to create a corpus. The corpus is now cleaned by removing stopwords, digits and punctuation marks. After cleaning the corpus, feature vectors are created with a dimension of 300.

- **Fitting different models**

For modelling, various regression algorithms are tried:

1. **Regularized Elastic net Linear Regression**
2. **Random Forest Regressor**
3. **Light Gradient Boosting Regressor**
4. **Catboost**
5. **XGboost**

- **Tuning the hyperparameters**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting of the data. Different Hyper parameter techniques like GridSearchCV, RandomizedSearchCV. The trained model is also cross validated by using KFold cross validation technique.

- **Model Interpretation**

Model interpretation is done by using SHAP value plot on a simple XGboost regression model which is taken as a base model. It is done to

determine the features that were most important while model building and the features that didn't put much weight on the performance of our model.

5. Algorithms used:

1. Regularized Elastic net

Linear Regression:

Elastic nets are the hybrid of both Ridge regression and the Lasso. As it handles the penalty of both the regression types discussed previously. Elastic nets linearly combine the penalties of the Lasso and Ridge (L1, L2). It generally gives good performance in case of large datasets but it can't always be true especially when the dataset is small. In such cases, the performance of the Elastic net may not be significant.

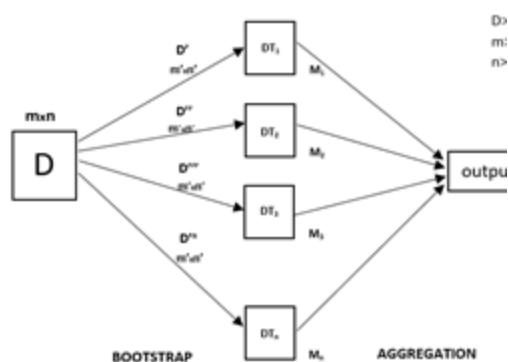
2. Random Forest Regressor:

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a

regression problem, the final output is the mean of all the outputs.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

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.



3. Light Gradient Boosting Classifier:

Gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

The base estimator for the Gradient Boost algorithm is fixed and i.e. *Decision Stump*.

Light GBM is a fast, distributed, high-performance gradient boosting framework based on a decision tree algorithm, used for ranking, classification and many other machine learning tasks. Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word 'Light'.

4. Catboost

CatBoost, short for Category Boosting, is an algorithm that is based on decision trees and gradient boosting like XGBoost, but with even better performance. CatBoost does especially well with data containing 'categorical variables'. In other models, categorical variables are handled through "OneHotEncoding" which creates additional columns to capture the information. Alternatively, CatBoost

starts by shuffling the data, creating “*permutations*”. For each, it assigns a “default” value for each class to the first few examples. Next, it calculates the value in each new row by looking at previous examples with the same class, and counting the number of positive labels, then performing a calculation. This captures additional valuable information, avoids sparsity, and speeds up computation. Then the model proceeds by building “*symmetric binary trees*” for each permutation of the data. To avoid overfitting, CatBoost builds new models at each step (n), by shuffling the rows and looking at n² previous examples.

4. XGboost:

Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling problems. 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.

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

6. Model performance:

Model can be evaluated by various metrics such as:

1. Mean squared error-

MSE or Mean Squared Error is one of the most preferred metrics for regression tasks. It is simply the average of the squared difference between the target value and the value predicted by the regression model.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

2. Root mean square error-

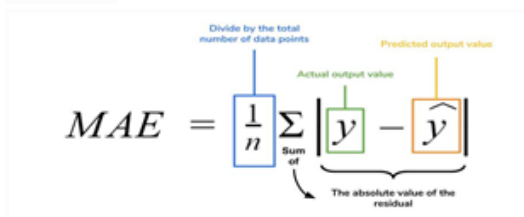
RMSE is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the value predicted by the model. It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$

3. Mean absolute error-

Mean Absolute Error is a model evaluation metric used with regression models. The mean absolute error of a model with

respect to a test set is the mean of the absolute values of the individual prediction errors on all instances in the test set. Each prediction error is the difference between the true value and the predicted value for the instance.



$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

4. R-squared-

Coefficient of Determination or R^2 is another metric used for evaluating the performance of a regression model.

The metric helps us to compare our current model with a constant baseline and tells us how much our model is better. The constant baseline is chosen by taking the mean of the data and drawing a line at the mean. R^2 is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R^2 will always be less than or equal to 1.

$$R^2 = 1 - \frac{MSE(\text{model})}{MSE(\text{baseline})}$$

5. Adjusted R-squared-

Adjusted R^2 depicts the same meaning as R^2 but is an improvement of it. R^2 suffers from the problem that the scores improve on increasing

terms even though the model is not improving which may misguide the researcher.

Adjusted R^2 is always lower than R^2 as it adjusts for the increasing predictors and only shows improvement if there is a real improvement.

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1} \right) \times (1 - R^2) \right]$$

where:

n = number of observations

k = number of independent variables

R_a^2 = adjusted R^2

7. 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.

In this project, Grid Search CV and Randomized Search CV are used for hyperparameter tuning.

1. Grid Search CV-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 biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

2. **Randomized Search CV-** In

Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control.

8. Results and Observations

Observations from EDA:

1. Most of the videos have less than 4 million views.
2. Most of the comments have less than 500 comments.
3. TED-Ed is the most frequent event of TED.
4. Speakers having 'writer' occupation are the most frequent speakers at the TED event.
5. There is an increase in the trend of videos released every year. But 2019 had the most video uploads with a

drastic decrease in 2020 because of **COVID - 19**.

6. The first quarter of the year sees most video uploads
7. Alex Gendler is the most frequent speaker as he acts as a host and narrator in many videos.
8. All videos are available in English(en) followed by Spanish - español(es).
9. Almost 50% of total videos are tagged under '**science and technology**'
10. Duration has no any kind of relationship with views.
11. When the videos first came out then it had the maximum average views, then a dip was observed in the following years followed by slight increase then again steady dip.

The dataset is now fitted into all the regression models suggested above. All the models are compared with the metric score 'R-squared'. Out of all of the models, the R2 score of Light GBM comes out to be the superior with the score of 86.1% and train set having score (88.8%). So both train and test set are almost comparable thus minimising the overfitting. We have also combined the score of different regression models using Stacking regressor and Voting regressor and both are giving the best results.

9. Conclusion:

In this project, we have used different classification algorithms to predict the possibility of purchasing vehicle insurance by health insurance buyers of insurance companies. First, the dataset is assessed and exploratory data analysis. Insights from EDA became the base of further data preprocessing and feature engineering. The dataset is now split into a train and test set as there should be no any information leakage from the test set. From the Model interpretation on the baseline model which is taken as a simple XGboost model, feature importance plots are found using SHAP interpretability. To obtain the best fit model, a metric score 'R-squared' for each model is compared. Among all the models, LightGBM gave the highest score after overcoming the overfit. As a result, it was possible to derive a predictive model with higher scores. Thus we were able to predict the views in each TED video.

References:

1. <https://www.kaggle.com/>
2. https://scikit-learn.org/stable/supervised_learning.html