TED Talk Views Prediction

Mayank Kumar
Sarvesh kumar Yadav
Hariom Bhardwaj
Shivam Mishra
Saifuddin Raja
Data science trainee,
AlmaBetter, Bangalore

Abstract:

TED is devoted to spreading powerful ideas on just about any topic. It is an exhibit for speakers introducing incredible, all-around shaped thoughts in less than 18 minutes.

Founded in 1984 by Richard Salman as a non-profit organization that aimed at bringing experts from the fields of Technology, Entertainment, and Design together, TED Conferences have gone on to become the largest pool 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.

1.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

Target Variable

Views: The number of views for each talk

2. Introduction

Ted Talks is one of the organisations that is always employing Machine Learning algorithms to increase the number of views the videos receive. They accomplish this by comprehending the significance of perspectives in conjunction with other critical provisions in order to increase the viewer's fulfilment by prescribing recordings based on standard perspectives that have been influenced by highlights such as subjects, comments, and so on.
The main objective is to build a predictive model, which could help in predicting the views of the videos uploaded on the TEDx website.

3. Steps involved:

- **Data Collection**

  We Mounted the drive and burdened the csv document into a data frame. Also, we had stacked our dataset that is given to us in .csv record into data frames.

- **Exploratory Data Analysis**

  We have Performed Exploratory data analysis where we looked for duplicate, missing and outliers' values in the given dataset. We have analysed and compared our target feature that is 'Views' with other independent variables. It gave us a clear picture of all important features with respect to 'Views'.

a. **Numerical Variables:**
   - Talk_id
   - Views
   - Comments
   - duration

b. **Textual Variables:**
   - Title
   - Speaker_1
   - Recorded_date
   - Published_date
   - Event
   - Native_lang
   - Url
   - Description

c. **Dictionaries:**
   - Speakers
   - Occupations
   - About_speakers
   - Related_talks

d. **List:**
   - topics

1. Continuous variable:

a. Views: The target variable is positively skewed.

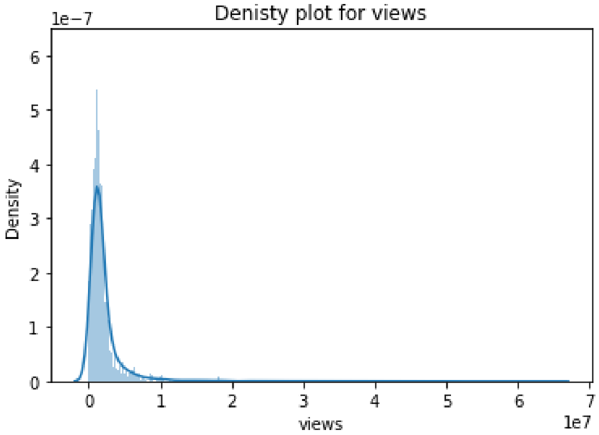
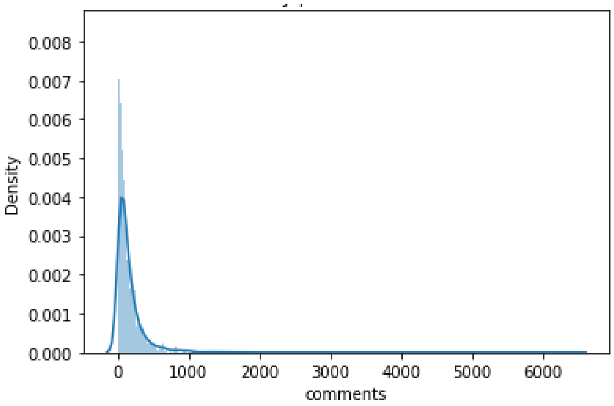


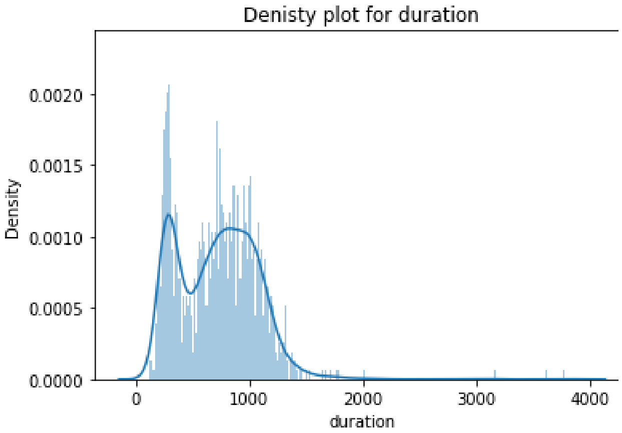
Fig: Density plot of views

The other continuous variables have following distributions:

b. Comments



c. Duration



All of the continuous data are skewed.

2. Missing value Detection and Treatment:

In our dataset, the max missing values in the Comments column i.e 16.35% of total records. This might incline to perturb our mean absolute score hence we have performed KNN nan value imputer for numerical features and superseded categorical features nan values with the value ‘Not Available’. We opted to impute nan values and not drop them due to the size of the data set

3. Feature Engineering

We have utilized domain cognizance to extract features (characteristics, properties, attributes) from raw data. A feature is a property shared by independent units on which analysis or presage is to be done.

- **<u>Outlier Detection and Treatment</u>**

  We have checked for outliers using boxplot, and have detected it in two variables:

  1.duration

  2. number of languages

  We have done outlier treatment by replacing outliers with mean values to prevent high errors that were influenced by outliers.

4. Experimenting with Models

We have tried various regression algorithms like: Decision tree, Random Forest, CatBoost, XGB Regressor,etc.

4.1. ML Algorithms:

1. Decision Tree Regression:

Decision Trees (DTs) are non-parametric supervised learning methods used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can verbalize that the purity of the node increases with deference to the target variable. The decision tree splits the nodes on all available variables and then culls the split which results in the most homogeneous sub-nodes.

Important Terminology related to Decision Trees:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
4. **Leaf / Terminal Node:** Nodes that do not split are called Leaf or Terminal nodes.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

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. XGBoost Regressor:

XGBoost is well known to provide better solutions than other machine learning algorithms. In fact, since its inception, it has become the "state-of-the-art" machine learning algorithm to deal with structured data.

- **Speed and performance** : Originally written in C++, it is comparatively faster than other ensemble classifiers.
- **Core algorithm is parallelizable** : Because the core XGBoost algorithm is parallelizable it can harness the power of multi-core computers. It is also parallelizable onto GPU’s and across networks of computers making it feasible to train on very large datasets as well.
- **Consistently outperforms other algorithm methods** : It has shown better performance on a variety of machine learning benchmark datasets.
- **Wide variety of tuning parameters** : XGBoost internally has parameters for cross-validation, regularization, user-defined objective functions, missing values, tree parameters, scikit-learn compatible API etc.

XGBoost (Extreme Gradient Boosting) belongs to a family of boosting algorithms and uses the gradient boosting (GBM) framework at its core. It is an optimized distributed gradient boosting library.

4. CatBoost Regressor:

CatBoost is a relatively new open-source machine learning algorithm, developed in 2017 by a company named Yandex. Yandex is a Russian counterpart to Google, working within search and information services. One of CatBoost’s core edges is its ability to integrate a variety of different data types, such as images, audio, or text features into one framework. But CatBoost also offers an idiosyncratic way of handling categorical data, requiring a minimum of categorical feature transformation, opposed to the majority of other machine learning algorithms, that cannot handle non-numeric values. From a feature engineering perspective, the transformation from a non-numeric state to numeric values can be a very non-trivial and tedious task, and CatBoost makes this step obsolete.

CatBoost builds upon the theory of decision trees and gradient boosting. The main idea of boosting is to sequentially combine many weak models (a model performing slightly better than random chance) and thus through greedy search create a strong competitive predictive model. Because gradient boosting fits the decision trees sequentially, the fitted trees will learn from the mistakes of former trees and hence reduce the errors. This process of adding a new function to existing ones is continued until the selected loss function is no longer minimized.

In the growing procedure of the decision trees, CatBoost does not follow similar gradient boosting models. Instead, CatBoost grows oblivious trees, which means that the trees are grown by imposing the rule that all nodes at the same level, test the same predictor with the same condition, and hence an index of a leaf can be calculated with bitwise operations. The oblivious tree procedure allows for a simple fitting scheme and efficiency on CPUs, while the tree structure operates as a regularization to find an optimal solution and avoid overfitting.

4.2. Hyperparameter 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 the 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.

We used Grid Search CV, Randomized Search CV, and Bayesian Optimization for hyperparameter tuning. This also results in cross-validation, and in our case, we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

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.

5. Natural language Processing:

One of the main reasons natural language processing is so critical to businesses is that it can be used to analyze large volumes of text data, like social media comments, customer support tickets, online reviews, news reports, and more.

All this business data contains a wealth of valuable insights, and NLP can quickly help businesses discover what those insights are.

It does this by helping machines make sense of human language in a faster, more accurate, and more consistent way than human agents.

We have 4 features that have the textual information in it.

1. title
2. occupations
3. topics
4. description

Steps followed for NLP

1. Convert all the words into its lower case.
2. Removed Punctuations
3. Removed Stopwords
4. Word Clouds
5. Split Data into Train and Test set.
6. Count Vectorization / TF-IDF vectorization
7. ML Model

6. Conclusion:

We reached the quit of our exercise. Starting with loading the statistics to date we've finished EDA , null values treatment, encoding of specific columns, characteristic choice after which version building. In all of those fashions our mistakes were withinside the variety of 2,00,000 that's round 10% of the common views. After hyper parameter tuning, we've averted overfitting and reduced mistakes through regularizing and decreasing getting to know rate. Given that handiest 10% is mistakes, our fashions have accomplished thoroughly on unseen statistics because of different factors like characteristic choice,accurate version choice,etc.

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
4. Median