Prediction and Analysis of Youtube trending videos using Machine Learning Methods

Sonali Ramchandra Kumbhar (L00163484), *M.Sc. Big data Analytics and Artificial Intelligence* ,
Department of Computing, *Letterkenny Institute of Technology*

***Abstract*—With the advent of social media and the internet, massive amounts of data are generated daily. But sifting through data to find insight takes time and effort. Many Big Data technologies, statistics, data mining, and machine learning algorithms are used by corporations and enterprises to improve their performance every day. In this study, we examine YouTube trending video data to determine what quickly catches the user's eye. Video sharing website YouTube allows users to upload, review, share, store, and even make their own videos. Videos that are now trending on YouTube are different from popular videos that have a certain number of likes and views. There is still a lot of work to be done on YouTube trending video analysis. To anticipate the amount of views for famous YouTube videos, the researchers utilized a Machine Learning model using linear regression. The study analyzes Random Forest, Linear Regression, and Naive Bayes classification, Adaboost, Isotonic Regression model methods for predicting how long a video would be trending from its upload time. In addition, we compare and monitor the fluctuation of activity over time in relation to the type of the event, which has an impact on the quality of our study. By doing a direct study, we are able to clearly identify the viewing patterns that pertain to each group independently.**

Keywords: Exploratory Data Analysis, Prediction of YouTube data, Python, Random forest, Data cleaning Linear Regression, Big data, Machine learning, Adaboost

I. Introduction

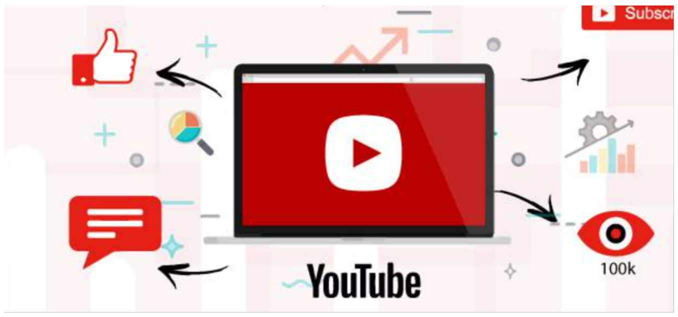


Fig. 1.

YOuTube is the world's most popular online video sharing platform. YouTube, which was created in 2005, has grown into a world in its own right. In fact, YouTube gets over 4 billion views every day (Iman Barjasteh et al., 2014). YouTube offers interactive video features for both the general public and content creators, such as Views, which displays the total number of people who have seen a video. A video's popularity is affected by the amount of views, and it takes time for a video to achieve popularity. There is usually something that catches people's attention fast and falls into YouTube's trending videos category. Trending videos are not now popular, but they may become so in the future. Despite their importance, famous YouTube videos have yet to be investigated. YouTube has become one of the most major commercial platforms, with over a billion unique monthly visitors and 72 hours of video uploaded every minute (Iman Barjasteh et al., 2014). In 2013, YouTube provided brand management, advertising, and marketing services. When a video becomes viral, it is made freely available to the public, attracting widespread attention. Because predicting which material will be popular in the near future is challenging, machine learning predictive analysis is offered. To apply aspiring to the information at hand, we use a range of approaches from the Scikit/Learn package.

II. Data Preparation

Data from various sources must be identified, cleaned, formatted, reorganized and Data may be cleaned by eliminating empty rows and replacing in missing information. The data is missing and may be added. In certain cases, there may be knowledge outliers. What we know may not be consistent. Data transformation improves data processing efficiency. There are different csv files in the dataset. Each csv file is categorized by country. First we will combine all these csv files in single dataframe. And video_id is the index. Column names in our dataset :

videoid, trending date, title, channeltitle, categoryid, publishtime, tags views, likes, dislikes, commentcount, thumbnaillink, commentsdisabled and description videoerrororremoved ratingsdisabled.

In this dataset we have added new column country. And our final dataframe name is 'combineddata'.

III. Data cleanup

For cleaning the data we have to remove null rows.

combined-data = combined-data.dropna() combined-data.info()

In below step we will removed the all non-numeric columns because we have processed them and stored the information captured in them dataset as numbers.

combineddata.drop(['trendingdate', 'title', 'channeltitle', 'category', 'publishtime', 'tags', 'views', 'likes', 'dislikes', 'commentcount','thumbnaillink', 'description', 'publishdate'], axis = 1,inplace = True)

IV. Methodology

Models: We have used four different models, a. Lasso regression b. Adabo c. Naive Bayes Classifier d. Isotonic Regression

A. Lasso regression

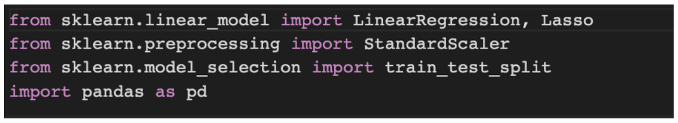
Lasso Regression

Using a lasso regression is a method of regularizing data. It is preferred over regression approaches because of its ability to provide more precise predictions. Shrinkage is a factor in this model. It is called "shrinkage" when data values are reduced to a single number, which is called the "mean." Simple, sparse models are encouraged by the lasso approach. Models with high degrees of multicollinearity or when specific portions of model selection, such as variable selection/parameter reduction, need to be automated benefit greatly from this sort of regression. Mathematical equation

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

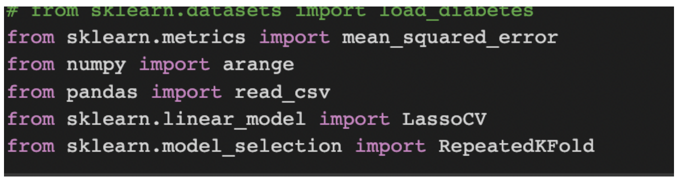
Fig. 2.

= amount of shrinkage. We have used below lasso regression libraries



```
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import pandas as pd
```

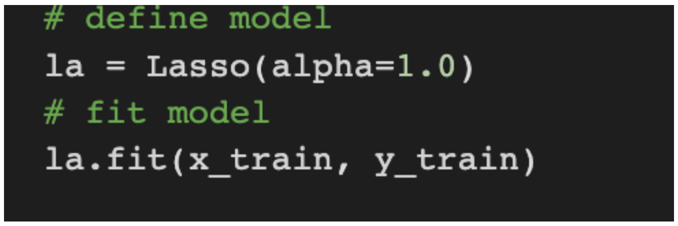
Fig. 3.



```
# from sklearn.datasets import load_diabetes
from sklearn.metrics import mean_squared_error
from numpy import arange
from pandas import read_csv
from sklearn.linear_model import LassoCV
from sklearn.model_selection import RepeatedKFold
```

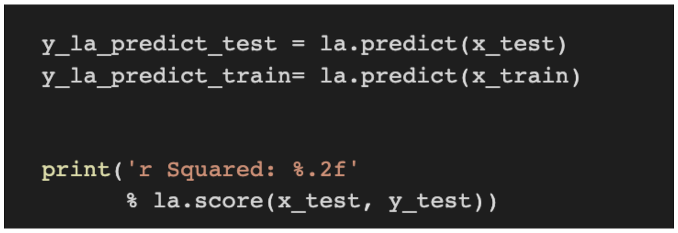
Fig. 4.

Then we have created the model and fit the model



```
# define model
la = Lasso(alpha=1.0)
# fit model
la.fit(x_train, y_train)
```

Fig. 5.



```
y_la_predict_test = la.predict(x_test)
y_la_predict_train= la.predict(x_train)


print('r Squared: %.2f'
      % la.score(x_test, y_test))
```

Fig. 6.

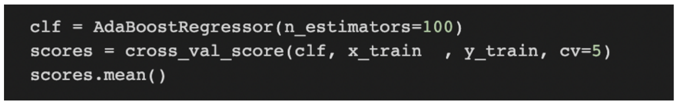
Classifying Predictors and Target

We have used cross validation in this method for calculate the r square.

B. Adaboost

AdaBoost is an algorithm that combines predictions from small, one-level decision trees known as decision stumps. The AdaBoost technique uses decision stump algorithms AdaBoost uses a lot of weak models for calculate the correct prediction. Training begins with a single decision tree, which is used to look for misclassified instances and to increase the weight given to such examples. The same data is used to train a new tree, but this time the weighting of misclassification mistakes is taken into account. Repetition of this procedure is necessary until the desired number of trees added.

In this project we have used cross validation in adaboost.



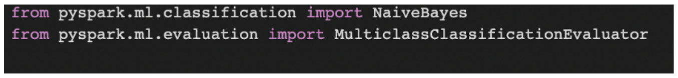
```
clf = AdaBoostRegressor(n_estimators=100)
scores = cross_val_score(clf, x_train  , y_train, cv=5)
scores.mean()
```

Fig. 7.

C. Naive Bayes Classifier

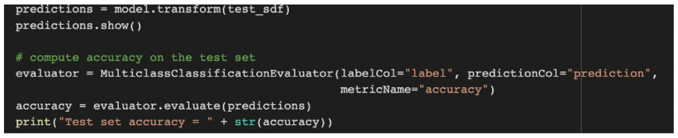
Naive Bayes is a simple multiclass classification algorithm with the assumption of independence between every pair of features. The training of Naive Bayes may be completed in a reasonably short period of time. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes' theorem to compute the conditional probability distribution of label given an observation and use it for prediction. MLlib supports multinomial naive Bayes and Bernoulli naive Bayes. For multinomial naive Bayes, each observation is a document, and each feature is a term, with the value of the term being the word's frequency. No negative feature values. The model type may be either multinomial or bernoulli, with multinomial being the default. Set the parameter to apply additive smoothing. Document categorization normally requires sparse feature vectors, thus provide sparse vectors as input. Because training data is only used once, caching is unnecessary.

In our notebook we have used below spark lib for naïve bayes classifier



```
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

Fig. 8.

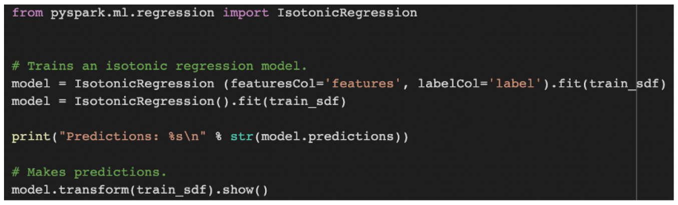


```
predictions = model.transform(test_sdf)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

Fig. 9.

D. *Isotonic regression*

Isotonic regression is a kind of regression method. Isotonic regression is a mathematical problem where we have finite set of real numbers Y=y1,y2,...,ynY=y1,y2,...,yn representing observed responses and X=x1,x2,...,xnX=x1,x2,...,xn the unknown response values to be fitted finding a function that minimises f(x)=i=1nwi(yixi)2(1)(1)f(x)=i=1nwi(yixi)2 with respect to complete order subject to x1x2...xnx1x2...xn where wiwi are positive weights. The outcome is isotonic regression, which is unique. It's a least squares problem with an order limitation. To put it simply, isotonic regression is a monotonic function. An IsotonicRegressionModel is returned after training, which may be used to predict labels for both known and unknown features. Isotonic regression's outcome is handled as a piecewise linear function. We have used isotonic spark libraries and model as below:



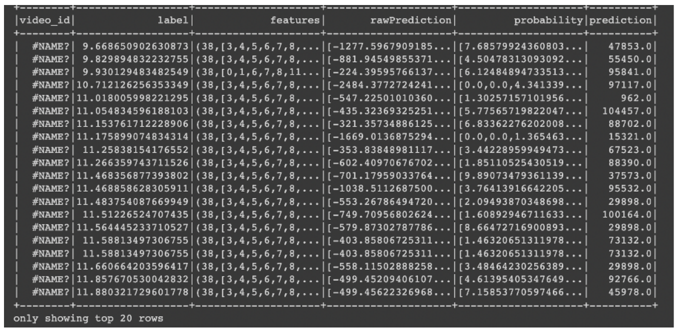
```
from pyspark.ml.regression import IsotonicRegression


# Trains an isotonic regression model.
model = IsotonicRegression (featuresCol='features', labelCol='label').fit(train_sdf)
model = IsotonicRegression().fit(train_sdf)

print("Predictions: %s\n" % str(model.predictions))

# Makes predictions.
model.transform(train_sdf).show()
```

Fig. 10.

V. Result

In our project we have used two regression models and one ensemble method and one classifier model. For accuracy we have used cross validation. Cross validation method produces performance evaluation matrix which includes accuracy, score and precision. Matrix provides expanded and knowledgable results. We have used lasso regression as linear method and Adaboost as ensemble method. Prediction and accuracy as below:

Model	Score	Prediction
AdaBoost	0.86	0.66
Lasso Regression	0.87	

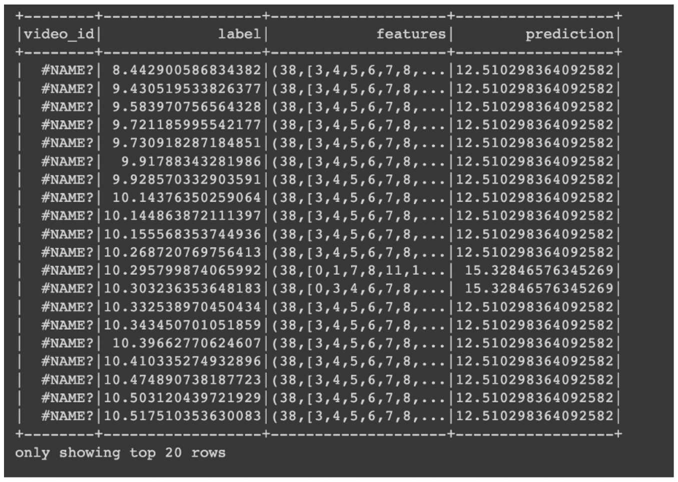


```
+--------+-----------------+--------------------+--------------------+--------------------+----------+
|video_id|            label|            features|       rawPrediction|         probability|prediction|
+--------+-----------------+--------------------+--------------------+--------------------+----------+
|  #NAME?| 9.668650902630873|(38,[3,4,5,6,7,8,...|[-1277.5967909185...|[7.68579924360803...|   47853.0|
|  #NAME?| 9.829894832232755|(38,[3,4,5,6,7,8,...|[-881.94549855371...|[4.50478313093092...|   55450.0|
|  #NAME?| 9.930129483482549|(38,[0,1,6,7,8,11...|[-224.39595766137...|[6.12484894733513...|   95841.0|
|  #NAME?|10.712126256353349|(38,[3,4,5,6,7,8,...|[-2484.3772724241...|[0.0,0.0,4.341339...|   97117.0|
|  #NAME?|11.018005998221295|(38,[3,4,5,6,7,8,...|[-547.22501010360...|[1.30257157101956...|     962.0|
|  #NAME?|11.054834596188103|(38,[3,4,5,6,7,8,...|[-435.32369325251...|[5.77565719822047...|  104457.0|
|  #NAME?|11.153761712228906|(38,[3,4,5,6,7,8,...|[-321.35734886125...|[6.83362276202008...|   88702.0|
|  #NAME?|11.175899074834314|(38,[3,4,5,6,7,8,...|[-1669.0136875294...|[0.0,0.0,1.365463...|   15321.0|
|  #NAME?| 11.25838154176552|(38,[3,4,5,6,7,8,...|[-353.83848981117...|[3.44228959949473...|   67523.0|
|  #NAME?|11.266359743711526|(38,[3,4,5,6,7,8,...|[-602.40970676702...|[1.85110525430519...|   88390.0|
|  #NAME?|11.468356877393802|(38,[3,4,5,6,7,8,...|[-701.17959033764...|[9.89073479361139...|   37573.0|
|  #NAME?|11.468858628305911|(38,[3,4,5,6,7,8,...|[-1038.5112687500...|[3.76413916642205...|   95532.0|
|  #NAME?|11.483754087669949|(38,[3,4,5,6,7,8,...|[-553.26786494720...|[2.09493870348698...|   29898.0|
|  #NAME?| 11.51226524707435|(38,[3,4,5,6,7,8,...|[-749.70956802624...|[1.60892946711633...|  100164.0|
|  #NAME?|11.564445233710527|(38,[3,4,5,6,7,8,...|[-579.87302787786...|[8.66472716900893...|   29898.0|
|  #NAME?| 11.58813497306755|(38,[3,4,5,6,7,8,...|[-403.85806725311...|[1.46320651311978...|   73132.0|
|  #NAME?| 11.58813497306755|(38,[3,4,5,6,7,8,...|[-403.85806725311...|[1.46320651311978...|   73132.0|
|  #NAME?|11.660664203596417|(38,[3,4,5,6,7,8,...|[-558.11502888258...|[3.48464230256389...|   29898.0|
|  #NAME?|11.857670530042832|(38,[3,4,5,6,7,8,...|[-499.45209406107...|[4.61395405347649...|   92766.0|
|  #NAME?|11.880321729601778|(38,[3,4,5,6,7,8,...|[-499.45622326968...|[7.15853770597466...|   45978.0|
+--------+-----------------+--------------------+--------------------+--------------------+----------+
only showing top 20 rows
```

Fig. 11.

And in Pyspark we have used Isotonic Regression method and Naïve Bayes classifier as classification method. Naïve Bayes classifier method prediction result as below:

Isotonic Regression method Prediction Result as below:



```
+--------+------------------+--------------------+------------------+
|video_id|             label|            features|        prediction|
+--------+------------------+--------------------+------------------+
|  #NAME?| 8.442900586834382|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 9.430519533826377|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 9.583970756564328|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 9.721185995542177|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 9.730918287184851|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|  9.91788343281986|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 9.928570332903591|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 10.14376350259064|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.144863872111397|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.155568353744936|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.268720769756413|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.295799874065992|(38,[0,1,7,8,11,1...| 15.32846576345269|
|  #NAME?|10.303236353648183|(38,[0,3,4,6,7,8,...| 15.32846576345269|
|  #NAME?|10.332538970450434|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.343450701051859|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?| 10.39662770624607|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.410335274932896|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.474890738187723|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.503120439721929|(38,[3,4,5,6,7,8,...|12.510298364092582|
|  #NAME?|10.517510353630083|(38,[3,4,5,6,7,8,...|12.510298364092582|
+--------+------------------+--------------------+------------------+
only showing top 20 rows
```

Fig. 12.

VI. Conclusion

Property taxes are an extremely crucial aspect of becoming a homeowner. Residents can choose to have their property taxes added to their mortgage bills, which the lender deposits in an escrow account, or they can pay them separately, but they must do so on a timely basis. Property taxes are assessed by governments depending on the location and value of the property. Municipalities and states utilise the money collected from homeowners' property taxes to fund important services and infrastructure, such as police and fire protection, schools, road and highway building, and a variety of other purposes that differ depending on the jurisdiction. Due to the continued rise in home prices, which translates into higher property taxes, homeowners must pay their property taxes on time. Failure to do so results in the local government imposing a tax lien on your property, which must be paid within a specified period, or else the property is foreclosed. Using Machine Learning and Artificial Intelligence can help in predicting the total of the Tax which each state will be giving and that can take as feedback o minimize. Using the exploratory data analysis, the important correlated features can be taken into the account for putting efforts in minimizing it to have a lesser tax with no legal issues.