Big Data Analytics using Machine Learning (YouTube Dataset)

Vineeth Menon, MSc Big Data Analytics, L00163249

***Abstract*— The second technical project report is based on application of Machine Learning supervised and unsupervised models on a youtube dataset that can provide insights on total likes, dislikes, comments, views etc. for specific categories of videos among 3 countries namely Canada, US and India. The provided task is to select an appropriate regression and ensemble models to predict 'total views' based on accuracy scores and other hyperparameter optimization techniques. A classification model is implemented on one of a categorical variable.Project is executed on Colab using Pandas, Scikit-Learn and Pyspark Mlib libraries.**

***Index Terms*— Regression, Classifiers, Gradient Boosting Regressor, Ensemble, Random Forest, Linear, Logistic Regression**

I. Introduction

THe tasks in second technical projects aims for implementation of regression models and classification models on a youtube dataset consisting of top trending videos. Top performers on the YouTube trending list are music videos, comedy, sports, science, celebrity and/or reality TV performances and other random viral videos. There are 3 CSV files representing trending videos of 3 different countries namely Canada, US and India.

In data loading and pre-processing, the CSV files are loaded using google.colab library. All 3 CSV files are merged into a single dataframe. A new column labelled as 'country' is introduced in the merged dataframe. The dataframe is named as 'combined_data'. This is used further for analysis and feature engineering.

The jupyter notebook provided for this project has covered exploratory data analysis. This includes studying the mean, standard deviation, minimum and maximum values of all numerical features. As the range of values were high, using Numpy libraries, log values of these features were calculated. Now the range and variance has been reduced making it suitable for unbiased analysis. Many steps have been done in feature engineering such as splitting up published time, converting categorical features in the dataset into one hot vectors and selecting labelCol and featureCol. Hence to summarise, all further analysis is focussed on total views in each video. The log value of the view count is treated as a 'label' and 38 columns are considered as 'features' for fitting into different models.

II. Regression Models

A regression model is a function that represents the connection between a dependent, or target variable and one or more independent variables. These models are implemented when the target values are numerical or continuous data. There are different types of regression models available such as Linear regression, Non Linear regression (Polynomial, Poisson etc.) and Tree Based regression models. There are many subcategories of each type. In this project, some of the regression models are implemented and evaluated using Google Colab and Apache Pyspark as platforms [1].

A. Linear Regression

This is the most common regression method. It is already implemented in the provided notebook, however it is essential to evaluate the accuracy scores (R-sq value) with other regressions. In this model, there is a need to estimate a function 'y' consisting of column label (y_train) with a weighted linear function over inputs with 38 features (x_train). The R-sq value for the model is 0.86 and RMSE value is 0.66. Inorder to

B. Lasso Regression (Task1)

The Lasso is a sub type of linear model that estimates sparse coefficients. This property leads to reduction in features with fewer non zero coefficients. Lasso is imported through sklearn.linear_model libraries. If linear regression indicates that a label is dependent on n features, Lasso will reduce the n to specify only most relevant features to be part of the model. It is very effective and used extensively in the field of compressed sensing. The alpha value in the model is the lasso alpha parameter. As alpha value increases, more coefficients of features are treated as zero. If alpha is zero, the coefficients are same as linear regression. The optimum value of alpha is taken between 0 to 1 and its selected through hyper parameter tuning [1].

This model gives an R-sq value of 0.84 and RMSE value of 0.72.

C. Ridge Regression (Task1)

Ridge regression overcomes the issues with Ordinary Least Squares by putting a penalty on the magnitude of the coefficients. The ridge coefficients minimize a penalized residual sum of squares. This example takes the value of alpha as 1. Further validations can also be done through cross validation and hypertuning techniques. Larger the values, greater the shrinkage of coefficient values [1].

This model gives an R-sq value of 0.86 and RMSE value of 0.66.

D. Stochastic Gradient Descent (Task1))

SGD is a reliable learning regression algorithm used when the training set is large. The stochastic process is subjected to the samples randomly picked at individual iteration. SGDRegressor function is imported through sklearn to fit linear models. During the fit, iteration and tol value (stopping criteria) are entered as per parameter tuning [3].

The RMSE value for the model is coming in range of 10 to the power of 7 that is unusually high even after applying scaling. Hence it needed to be ruled out

III. ENSEMBLE METHODS

Ensemble methods is a machine learning technique that combines numerous base models in order to produce one optimal predictive model [4].

A. Random Forest

Random forest is the common and widely used ensemble method. It is an ensemble of multiple decision tree algorithms. The final predicted value is based on aggregated results. Random forest is used over decision trees to avoid overfitting which is a primary concern in decision tree algorithm.

The model is already implemented in the given notebook with parameter hyper tuning. The hyper tuning is done through GridSearchCV. Another method explored is RandomizedSearchCV. The accuracy for the model is 94%.

B. Gradient Boosting Regressor (Task2 and Task1)

Gradient Boosting Regressor (GBR) is a type of ensemble learning which is built on principle of Boosting methods unlike Random Forest that relies on Bagging methods. It gives a prediction in the form of an ensemble of weak decision tree prediction models[2].

Like random forest, GBR needs tuning for various parameters such as n_estimators that determines number of boosting stages to perform. Gradient boosting is robust to over-fitting; hence a large number can lead to better performance. For this algorithm, the value starts from 200 to all the way to 2000. max_features are 3 types namely ‘auto’, ‘sqrt’ and ‘log2’. This helps to choose the number of features fully or square root value of it. Max_depth defines maximum depth needed for an individual regression estimator. It decides the number of nodes in a tree. It is a vital parameter to tune. With all these tuning, one can select the best parameter combination for best accuracy result. In this notebook, tuning is done through RandomizedSearchCV [1].

The accuracy for GBR model is 74% with features from Dimensionality Reduction using Principal Component Analysis. This algorithm has been implemented using Scikit learn as well as Mlib library.

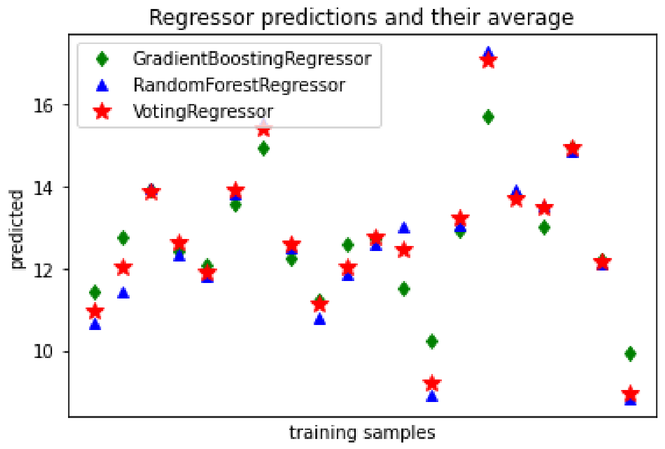

Fig. 1. Comparison of predicted values among 3 ensemble regressors

C. Voting Regressor (Task1)

A voting regressor is an ensemble method that fits several base regressors implemented on the whole dataset. At that point it averages the individual predictions to form a final prediction. It is useful when one has to choose between a set of well performing ensemble models and balancing out their individual weaknesses [1]. The accuracy for Votor Regressor turned out to be 90%. This is a good accuracy for an ensemble method. The fig.1 displays comparison of predicted values against each ensemble regressors.

The table represents the comaprison of all regressions used to predict the log views count based on input features.

Model	Accuracy Score	RMSE value	MSE value
Linear Regression	86%	0.667	0.45
Lasso Regression	84%	0.72	0.55
Ridge Regression	86%	0.67	0.5
Stochastic Gradient Descent	-135% (Neglected)	NA	NA
Random Forest	94%	0.44	0.19
Gradient Boosting Regressor	74%	0.91	0.82
Voting Regressor	91%	0.54	0.29

Fig. 2. Comparison of predicted values among 3 ensemble regressors (values will change slightly in each run)

[1]

IV. CLASSIFICATION MODELS

Classification models are implemented when the target values are categorical data (Binomial and Multinomial).

A. Logistic Regression (Task2)

Logistic regression is a linear model for classification instead of regression. Logistic regression is also known in the

[1]Github link: `https://github.com/Vineeth08/BDA--Technical-Project-2.git`

literature as “logit regression”, “maximum-entropy classification” or the “log-linear classifier” [1].

This is executed in spark.ml using mlib library. LogisticRegression function is imported from pyspark.ml.classification. For implementation, the label selected is ‘ratings_disabled’ for predicting the impact on ratings disabled variable due to all features. There are 2 values in label column as ‘true’ and ‘false’. For executing the classifier function, ‘ratings_disabled’ is converted into indexed values using pyspark.sql.types library followed by feature engineering using VectorAssembler and pipeline creation. The new dataframe is saved as modified_data_sdf is splitted into training and testing (80:20) and entered into the function. Hyperparameter tuning is done for multiple values such as Elastic Net Mixing Parameters, Iterations and Logistic Regression params. These arrays are converted to list which will be called during LogisticRegression() model [2].

Range of accuracy scores are available using BinaryClassificationEvaluator with 99.5% being the best for 10 iterations, 0.0 elastic net param and 0.1 regParam values. areaUnderROC is calculated by obtaining the receiver-operating characteristic as a dataframe. Calculated area under ROC curve = 0.5.

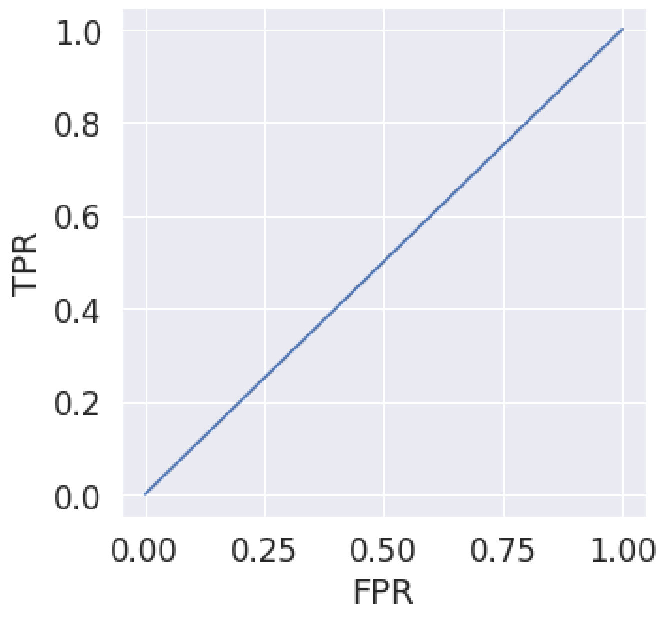

Fig. 3. TPR vs FPR graph (ROC curve with random performance level. AUROC = 0.5)

The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. The graph plot is displayed in Fig.3 Two areas separated by this ROC curve indicates a simple estimation of the performance level. ROC curves in the area with the top left corner (0.0, 1.0) indicate good performance levels, whereas ROC curves in the other area with the bottom right corner (1.0, 0.0) indicate poor performance levels [5].

[2]

V. CONCLUSION

The aim of the project is to understand the applications of machine learning models and multiple scenarios where it can be used for best outcomes. In task 1, linear regression is already implemented but it is needed to check the R-sq values and RMSE scores using other regressions because linear regression can be a victim of overfitting due to factor of multicollinearity. The project compares several regression techniques (linear and ensemble) with the accuracy scores and other measuring parameters.

Random Forest turns out to be the ensemble method that gives highest accuracy and least mean root square error values while predicting view_logs values. Votor regressor also provided a good accuracy level. In classification model, Logistic Regression methods provided a high accuracy rate for predicting if the ratings are able or disabled for a particular video.

The report and code files used Scikit-Learn and Apache Spark web pages as references to high extend along with lecture notes.

REFERENCES

[1] https://scikit-learn.org/stable/supervised_learning.htmlsupervised-learning n.d.
[2] https://spark.apache.org/docs/latest/ml-classification-regression.html n.d.
[3] Bottou, Léon. ”Stochastic gradient descent tricks.” In Neural networks: Tricks of the trade, pp. 421-436. Springer, Berlin, Heidelberg, 2012. – SGD
[4] https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f n.d.
[5] https://classeval.wordpress.com/introduction/introduction-to-the-roc-receiver-operating-characteristics-plot n.d.

[2]Github link: `https://github.com/Vineeth08/BDA--Technical-Project-2.git`