Big Data Analytics using Machine Learning

Mallikharjuna Rao S, MSc in Big Data Analytics and Artificial Intelligence, L00157129

***Abstract*— This technical report will discuss the accuracy of Linear regression, ensambled regression, and classification models of ML libraries Spark mllib, Sklearn on a large data set containing popular YouTube videos information. Furthermore, the reasons for selecting each ML model are discussed in detail. Finally, accuracy details are compared in order to form an opinion about the models as well as both ML libraries.**

***Index Terms*— Machine Learning, Big Data, Regression, Classification, Spark mllib, Sklearn**

I. Problem Statement

Using regression and classification machine learning models from sklearn and spark MLib to investigate the patterns of popular videos on the video sharing platform YouTube from a data set containing a daily record of the top trending YouTube videos. Additionally, tuning the ML models and visualizing the accuracy details.

II. Linear Regression using sklearn

Ridge linear regression, Lasso linear regression, and TweedieRegressor are used to solve the problem of predicting the views log using the rest of the features related to comments, ratings, likes and dislikes, and a few other related features.

A. Ridge linear regression

Ridge regression employs L2 regularization, which gradually reduces the weights to zero for each iteration. The goal of L2 regularization is to achieve simplicity. Because there isn't much of a linear relationship between the independent variables, multicollinearity is obvious, so Ridge linear regression is used here. Ridge linear regression introduces some bias into the predictions in order to bring the loss values, which the regular least squares method does not. [1]

B. Lasso linear regression

Ridge regression employs L1 regularization, also known as sparsity regularization. For each iteration, features that are less relevant and have a lower coefficient with the label are reduced. The goal of using Lasso regression is to make irrelevant features insignificant. [2]

C. TweedieRegressor

Tweedie regression models are ideal for dealing with data that is heavily skewed to the right and consists of continuous types of data. TweedieRegressor models are a better option than Logistic regression because the given data is not a perfect fit for the binomial distribution. Because the given data values for total view, which is the label in the problem, align with the Poisson distribution, the TweedieRegressor model is used to perform linear regression.

D. Gradient Boosting Regressor

Ensemble learning refers to the process of combining multiple models to improve performance. Gradient Boosting Regressor is implemented with the goal of outperforming individual models in terms of accuracy. In each iteration, GB optimizes differentiable loss functions for each regression tree.

E. Model Tuning

GridSearchCV from the sklearn model selection package is used to perform cross validation for multiple folds in order to find the best parameter values for all sklearn regression models.

The following are the best hyper parameter values for each regression:
Ridge : (alpha=10, max_iter=500)
Lasso : (alpha=0.1, max_iter=500, selection='random')
TweedieRegressor : (alpha=0.1, link='log', power=0)
Gradient Boosting : (max_depth: 25, n_estimators: 70).

F. Linear regression Results - sklearn

The table below depicts various accuracy scores for the aforementioned models (*MSE* - mean squared error and *RMSE* - Root mean square error). It is very evident that Gradient Boosting being an ensamble methods, clearly outperformed the other models in terms of accuracy.

Figures 1, 2, 3, and 4 show the prediction error details for the Ridge, Lasso, Tweedie, and Gradient Boosting regression methods.

Metric	Ridge	Lasso	Tweedie	GradientBoosting
R^2	0.865	0.857	0.86	0.91
MSE Test	0.45	0.53	0.47	0.28
MSE Train	0.44	0.53	0.47	0.00
RMSE	0.67	0.72	0.69	0.53

III. Linear Regression using Spark

A. Decision Tree Regressor

Decision Tree Regressor divides data into subsets before forming a tree to perform regression and predict the target

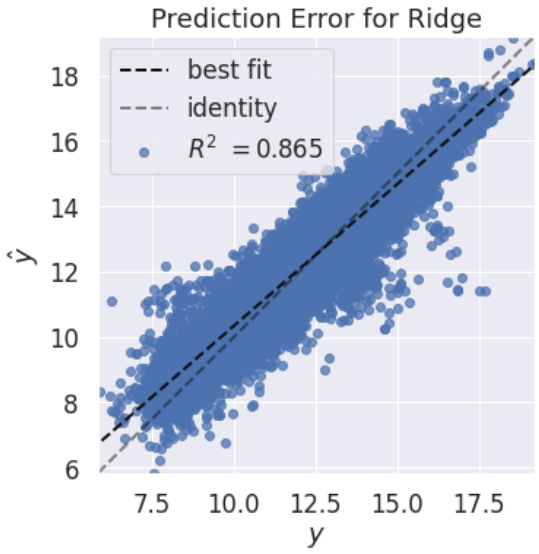

Fig. 1. Visualization Ridge linear regression

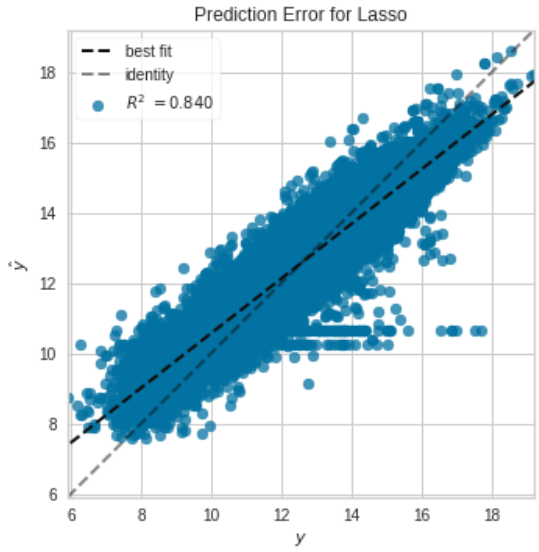

Fig. 2. Visualization Lasso linear regression

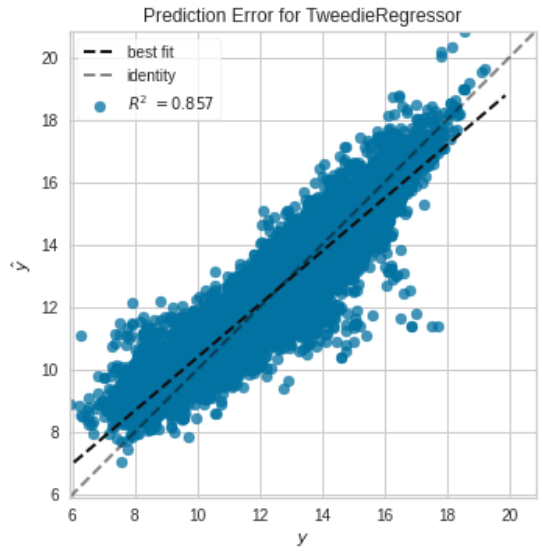

Fig. 3. Visualization Tweedie Regressor linear regression

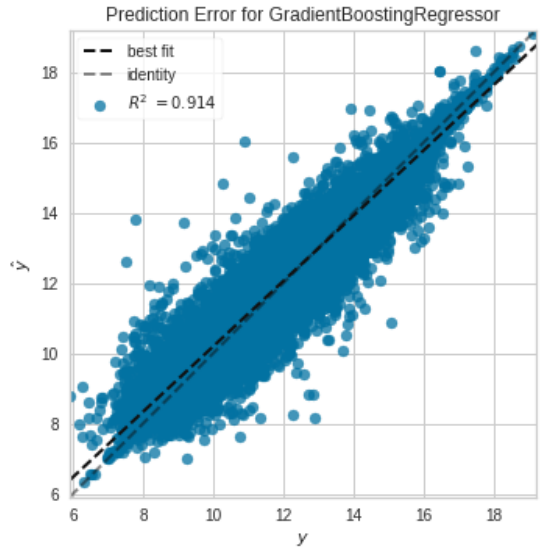

Fig. 4. Visualization Gradient Boosting linear regression

value. The reason for using this model is that the decision tree model ignores entropy, and the final predicted value is the average of the values of the feature variables of the node tree at that given position.

B. Gradient boosted tree regression

Gradient boosted tree regression is an ensamble model made up of several decision trees. The reason for selecting this model is that the loss will be reduced and performance will improve as a result of the multiple samples from different trees.

C. Spark Models Tuning :

Cross validation is implemented using ParamGridBuilder and CrossValidator from *pysparkml* tuning to determine the max depth and max bin values for decision trees in both (Decision tree and Gradient Boosted) regression models of spark.

D. Linear Regression Results - Spark

The accuracy details for Spark MLib's Decision tree and gradient boosting ML models are shown in the table below. Figures 5 and 6 portray the accuracy, MAE, and RMSE details for Spark MLib's decision tree and Gradient Boosting regression methods. It is possible to conclude that gradient boosting produces better results than Decision Tree.

Metric	DecisionTree	Gradient Boosted
R^2	.84	0.88
MAE	.53	0.46
RMSE	0.70	0.62

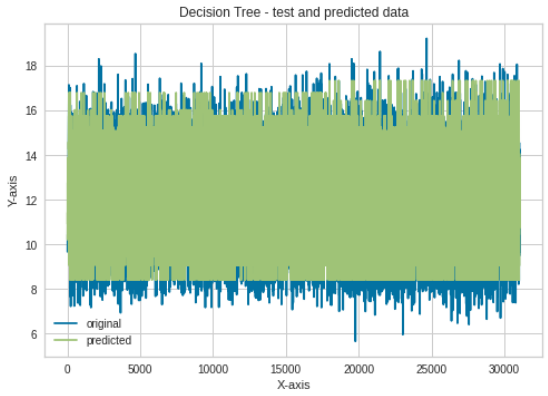

Fig. 5. Visualization Spark Decision Tree regression

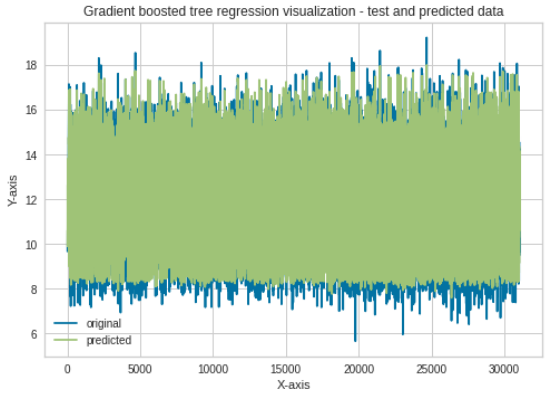

Fig. 6. Visualization Spark Gradient Boosting regression

IV. Classification using Spark

A. Data Preparation

A new prediction label is chosen to implement the classification on the given data set. View log is ideal for regression problems because it has continuous data values. As a result, the category id with categorical data is chosen for classification.

B. Decision Tree classification

Decision trees classification models outperform Decision trees regression models. The top down and greedy search approach followed by the model through the branches without backtracking is one of the reasons for choosing this model.

C. RandomForestClassifier

Random Forest Classifier is an ensamble method that combines multiple decision tree values. The objective of implementing this model is to reduce loss by using a relatively large number of trees.

D. Spark Models Tuning

To tune the model with the best set of parameter values, a param grid is built with ParamGridBuilder and cross validation is performed on the models with param grid values.

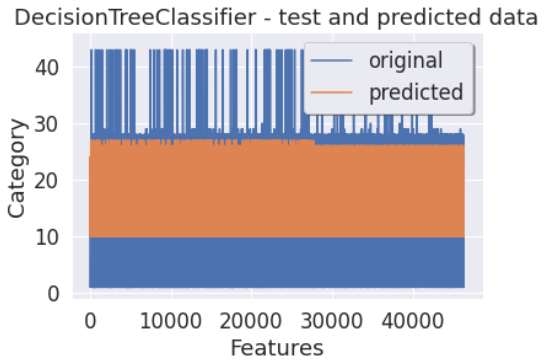

Fig. 7. Decision Tree classification - Visualization of test and predict data

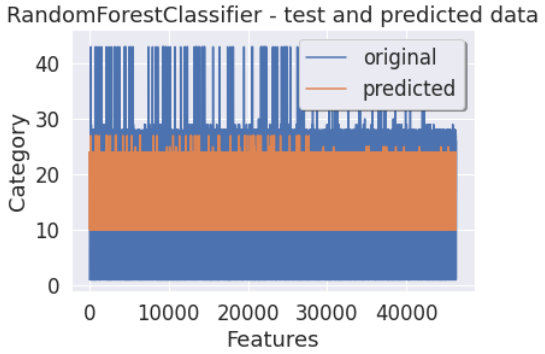

Fig. 8. Random Forest classification - Visualization of test and predict data

E. Results

The accuracy details for both classification models are shown in the table below. Though the accuracy values appear similar, random forest performed better with predictions after examining the visualization of the predictions, and random forest's performance will improve with more trees and iterations.

Furthermore, the accuracy scores for classification are low because dataset features are not intended for classification. Using "Natural Language Methods" to determine the "category" or "Catregory id" for the columns "Tags" and "Comments" would have resulted in better classification results.

Metric	Decision Tree	Random Forest
Accuracy	0.36	0.36
Test Error	0.64	0.64

V. Discussion

Based on the observations, it is possible to conclude that spark outperforms sklearn in terms of achieving better results and turnaround time. Sklearn performs in-memory processing in a non-distributed manner. As a result, it is best suited for working on small or medium-sized data bases. Spark's ML Lib, on the other hand, is the best choice for working in a distributed environment with large data sets. Sklearn, on the other hand, has more visualization options because it supports pandas and Matplotlib.

VI. GitHub Link

https://github.com/MallikharjunaSakhamuri/BDAwithML/tree/main

References

[1] Ridge Regression. (n.d.). [online] Available at: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf.

[2] Yıldırım, S. (2020). L1 and L2 Regularization — Explained. [online] Medium. Available at: https://towardsdatascience.com/l1-and-l2-regularization-explained-874c3b03f668 [Accessed 27 Feb. 2022].