Comparative Analysis of Machine Learning Algorithms for Emotion Detection

Mrunal Sawant[#1], Shrinivas Joshi[#2], Shailesh Haldankar[#3], Sanjana Dias[#4], Pradnya Kholkar[#5], Shaunak Pai Kane[#6]

[#1,#2]*Assistant Professor, Electronics and Communication Engneering Department, Agnel Institute of Technology and Design, Assagaon – Goa, India*

[,#3,#4,#5,#6] *Student, Electronics and Communication Engneering Department, Agnel Institute of Technology and Design, Assagaon – Goa, India*

Abstract

The Research paper focuses on comparison of various machine algorithms like Logistic Regression, Decision Tree, XGBoost (Extra Grad ient Boosting) and Convolutional Neural Network to determine Emotion of an Individual. All Algorithms were tested on Database consisting of images. The best emotion detection Algorithm is then used to determine the Stress of an Individual using Random Forest Algorithm.

Keywords - *Logistic Regression, Decision Tree, XGBoost and Convolutional Neural Network, Random Forest and Classifiers*

I. INTRODUCTION

Area of focus of this project is to do comparative analysis of various machine learning Algorithms and select the best Algorithm for determining the emotion of an Individual and hence determine Stress [1].

We have compared 4 Algorithms namely Logistic Regression, Decision Tree, XGBoost and Convolutional Neural Network. After determining the best algorithm for emotion detection, Random Forest Algorithm is used to determine the Stress of an Individual.

II. RELATED WORK

A number of methods are used to determine the stress of an individual. One methods detects the stress using EEG signals but is highly disadvantageous due to its high cost. Studies employ physiological data such as electro-dermal, muscular and cardiovascular activity to measure individual's stress. Other instruments include questionnaires which includes series of questions asked by psychiatrists can be used to estimate a person's stress. Measurement of galvanic skin response (GSR) using a GSR sensor system to further detect stress, however this method is not accurate. In our project we determine whether an individual is stressed or not using images of the students using the concept image processing and Machine Learning Algorithms. The proposed method does not use any kind of invasive methods or sensors as used in the other methods. The entire Project is divides into two sections wherein first Algorithm only determines the emotion of an individual over a period of time and the second Algorithm determines the stress level based on the emotion detected by the first Algorithm.

III. ALGORITHMS

A. *Logistic Regression*

Logistic regression is a technique borrowed by machine learning from the field of statistics. It is used to predict the probability of an outcome and is popularly used for classification tasks. The algorithm fits the given data to a logistic function and predicts whether an event will occur.

To generalise logistic regression to multiclass problems i.e. with more than two possible discrete outcomes we use multinomial logistic regression. Standard logistic regression is binomial and assumes two output classes whereas Multiclass or multinomial logistic regression assumes three or more output classes.

Function of Logistic Regression:

sklearn.linear_model.LogisticRegression(penalty='l2', dual=False, tol=0.0001, C=0.001, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='warn', max_iter=100, multi_class='warn', verbose=10, warm_start=False, n_jobs=-1,scoring ='Neg_log_loss',cv=2)

Attributes	Details	Value
Verbose	For the liblinear and lbfgs solvers set verbose to any positive number for verbosity.	10
Cv	Cross Validation	2

N_jobs	Number of CPU cores used when parallelizing over the classes. In this case we use all CPU cores	-1
Scoring	Type of Loss	Neg_log_loss
C	Inverse of regularization strength; must be a positive float. Smaller values specify stronger regularization.	0.001

B. Decision Tree

Decision Tree builds classes or regression models in the form of tree like structures. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. It splits the given data into smaller and smaller subsets till it attains a pure subset. While doing so a tree like structure is developed .The final tree constitutes of nodes and leaf nodes where nodes have two or more branches (other attributes) and leaf nodes represent the final decision (outcome). The best predictor is the root node and is the topmost node.

Function of Decision Tree:

DecisionTreeClassifier(*criterion='gini', splitter='best', max_depth=3,* scoring='neg_log_loss', *min_samples_split=2,* n_jobs=-1, *min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False, cv=2*)

Attributes	Details	Values
max_depth	The maximum depth of the tree.	3
Cv	Cross Validation	2

Scoring	Type of Loss	Neg_log_loss
N_jobs	Number of CPU cores used when parallelizing over the classes. In this case we use all CPU cores	-1

C. Convolutional Neural Networks

In deep learning, a convolutional neural networks is most commonly applied to analysing visual imagery. CNNs use a variation of multilayer perceptrons which require minimal processing. Convolutional Neural Networks are designed to process data through multiple layers of arrays. They are used in applications like face recognition or image recognition. The primary difference between CNN and any other ordinary neural network is that CNN takes input as a two-dimensional array and operates directly on the images rather than focusing on feature extraction which other neural networks focus on. The dominant approach of CNN includes solutions for problems of recognition.

Convolution Neural Network have following layers:

- Convolution
- ReLU Layr
- Pooling
- Fully connected

Function of CNN:

top_layer_model.compile(loss='categorical_crossentropy', optimizer=adamax, metrics=['accuracy'])

top_layer_model.fit(x_train_feature_map, y_train, validation_data=(x_train_feature_map, y_train), nb_epoch=FLAGS.n_epochs,batch_size=FLAGS.batch_size)

D. XGBoost

The XGBoost algorithm implements the gradient boosting decision tree algorithm. Boosting refers to a family of algorithms which converts weak learner to strong learners by adding new models to correct the errors made by existing models. Example is the AdaBoost algorithm that weights the data points that are hard to predict.

Gradient boosting is an approach where new models are created that predict the residuals or errors of their predecessors and then summed together to make the final prediction. It is called gradient

boosting since it uses a gradient descent algorithm to minimize the loss while new models are added.

$$L^{\zeta(t)}(q) = -\frac{1}{2}\{\sum_{j=1}^{T}((\sum_{i\in J} g_i)^2/\sum_{i\in I} h_i + \lambda) + \gamma T\}$$

Function of XGBoost:

XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bytree=1, gamma=1, learning_rate=0.02, max_delta_step=0, max_depth=5, missing=None, n_estimators=150, n_jobs=1, nthread=1, objective='multi:softprob', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=True, subsample=1)

Attributes	Details	Values
Learning Rate	Learning rate shrinks the contribution of each tree. There is a trade-off between learning rate and n_estimators.	0.02
N_estimator	It denotes the number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance.	150
Gamma	Lagrangian Multiplier for complexity control	1
Max_depth	Maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree.	5

IV. COMPARISON

Algorithm	Train Loss	Test Loss	Accuracy
DECISION TREE	1.89561262	1.90414399	19.33%
CNN	0.0048	6.51	-
LOGISTIC	0.93325895	1.35808188	50.52%
XGBOOST	0.06967287	0.06967485	99.71%

All the Algorithms were trained using the database. Hyper parameters of all the algorithms were chosen so as to get highest accuracy and minimize the difference between Train loss and Test loss. Since the images of all the emotions were imbalance in the database, they were balanced to avoid the problems of overfitting and under fitting the model. It was found that XGboost method gives highest accuracy for our project. Therefore this method will be used later on to determine stress of an individual [1]. Hyper parameters of Decision Tree, Logistic Regression and XGBoost were perfectly tuned in order to get highest accuracy and minimize the difference between Test Loss and Train Loss. However Hyper parameters were not perfectly tuned.

V. FUTURE WORK

Although XGBoost Algorithm gives best accuracy, Hyper parameters of CNN can be tuned to achieve high accuracy. To further increase the accuracy, Dimensional Reduction methods can be used to reduce the dimensions so that model will be trained and very high accuracy can be obtained. Dimensional Reduction methods removes unnecessary attributes which does not give significant increase in accuracy.

VI. CONCLUSION

After training all the Algorithms it was found that XGBoost Algorithm gives highest accuracy. Since accuracy of XGBoost is 99.7%, the accuracy of the Stress Detection will also be high. Thus by using XGBoost Algorithm, we can determine the emotions correctly and later can predict whether an individual is stressed more accurately using Random forest Algorithm [1].

REFERENCE

[1] Implementation of Stress Detection System by Mrunal Sawant, Shaunak Pai Kane, Shailesh Haldankar, Sanjana Dias and Pradnya Prabhu Kholkar, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN (P): 2249-6831; ISSN (E): 2249-7943 Vol. 9, Issue 1, Jun 2019, 11-14.

[2] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.