# Comparative analysis of various classification Machine Learning Algorithm on Driver Drowsiness Dataset

DR. UDAI BHAN TRIVEDI
*PSIT, Kanpur*

***Abstract— Real Time Drowsiness behaviours of a driver related to fatigue are in the form of eye closing, head nodding. The focus of this research paper is on the detection of blinks of eye by estimating the EAR (Eye aspect Ratio). This is achieved by monitoring the eyes of the driver throughout the entire video sequence. An IR camera will be used for capturing live video of driver eyes in all light conditions and frames will extracted for image processing scheme of video capturing. The Various Binary classifying algorithm will be applied on Driver Drossiness Dataset and feature like sagging leaning of driver's head and open/closed state of eyes will determine the state of the driver. The focus of this research paper is to classify the driver Drowsiness into two classes: 1. Open/Alert 2. Close/ Drowsiness through various classifying machine learning algorithms and determine the best performing algorithm for this purpose.***

***Indexed Terms-- Classification, Machine Learning, Driver Drowsiness, EAR (Eye Aspect Ratio)***

## I. INTRODUCTION

The drowsiness of the driver and rash driving are the major causes of road accidents, which result in loss of valuable life, and deteriorate the safety in the road traffic. Reliable and precise driver drowsiness systems are required to prevent road accidents and to improve road traffic safety. Various driver drowsiness detection systems have been designed with different technologies which have an affinity towards the unique parameter of detecting the drowsiness of the driver [1].

The Paper will compare various Machine learning classifying algorithm to train and test the model on Driver Drowsiness Dataset (https://www.kaggle.com/serenaraju/yawn-eye-dataset-new). The Research paper deals with automatic driver drowsiness detection based on visual information. The Algorithm will locate, track, and analyse both the drivers face and eyes, a scientifically supported measure of drowsiness associated with slow eye closure [2].

scope of this research paper is binary classification family of models/Algorithm which is selected as part of predictive modelling.

The aim of this research is to comparative analysis of various classification ML algorithm and find out the best classification algorithm to determine the Drowsiness of driver. The research paper will compare different Classification algorithm on the basis of Accuracy parameter. In this paper we compare following machine learning classifying algorithms from Sklearn [11]:

- Support Vector Machine
- Gaussian Naive Bayes
- Decision Trees
- Random Forest
- K Nearest Neighbors
- Stochastic Gradient Descent
- CNN

- The Support Vector Machine (SVM) is a classifier using a separating hyperplane as its formal definition. SVM creates a hyperplane that is used to divide distinct classes in multidimensional space by minimising error. The goal of SVM is to determine the optimum maximum marginal hyperplane for dividing the dataset into n classes. It has a high accuracy compared to other classifiers like logistic regression and decision trees, but it uses a lot of memory and is difficult to modify the parameter [12].

- Gaussian Naive Bayes is a Naive Bayes variation that allows continuous data and follows the Gaussian normal distribution. The Bayes theorem provides the basis for a collection of supervised machine learning classification algorithms known as Naive Bayes. It's a basic categorization approach with a lot of power.
- Decision Tree- When we want the output to be intuitive and the findings to be explainable to lay people with no technical experience and other stakeholders, the decision tree model is the top choice. If no linear link exists, the tree fails, and if not tweaked, it tends to overfit.
- Random forests- are a classification system that combines decision trees with random forests. Individual models provide predictions that are unrelated to one another in this situation. The result is a class that represents the mode of the individual trees' classes (classification). It solves the problem of overfitting in decision trees, but because the bootstrap sample selection is arbitrary, any combination of rows and features might cause bias in the tree and outcome.
- K-Nearest Neighbours: It makes predictions based on a data point's closest neighbours. As the model understands the local instance, there is a lack of global model interpretability. This method is interpretable if there are few characteristics; otherwise, it is not a suitable choice [13].
- Stochastic Gradient Descent- Overall, by lowering the loss function in gradient boosting, we improve the performance of the Stochastic Gradient Descent.
- The Convolutional Neural Network (CNN) is a type of neural network that (CNN) To train and evaluate deep learning CNN models, each input picture will be sent through a succession of convolution layers using filters (Kernels), Pooling, fully connected layers (FC), and the SoftMax function to identify an item with probabilistic values ranging from 0 to 1.

## II. RELATED RESEARCH

The list of paper has been revied in order to get an understanding of Driver drowsiness feature and how various ML algorithm will perform the classification.

| Author's Name/ Topic of Research Paper/ Journal/Conferences& Year of Publication | Discussion in the Paper |
|---|---|
| *Tianyi Hong, Huabiao Qin, & Qianshu Sun./ An Improved Real Time Eye State Identification System in Driver Drowsiness Detection/ IEEE International Conference on Control and Automation/2007* | In this paper the face region is detected using an optimised Haar-like feature detection scheme; the eye region is obtained by applying horizontal projection of the detected face and geometrical position of the eye on the face; and finally, the eye state is identified using a new complexity function with dynamic threshold. |
| *Tripathi, D.P., Rath, N.P./ A novel approach to solve drowsy driver problem by using eye-localization technique using CHT/ International Journal of Recent Trends in Engineering/2009* | This paper discussed an innovative way to alerting a motorist who tends to doze off while driving has been developed. A picture is obtained using this technique, which employs a tiny camera aimed straight at the driver's face. The skin region, i.e., the face, is segmented out of that image using the YCbCr colour space. Finally, the circular Hough transform is used to detect whether the eyes are open or closed, allowing for eye localisation. If the driver's eyes are closed for four frames in a row, the system concludes that he or she is becoming asleep and emits a warning signal. |

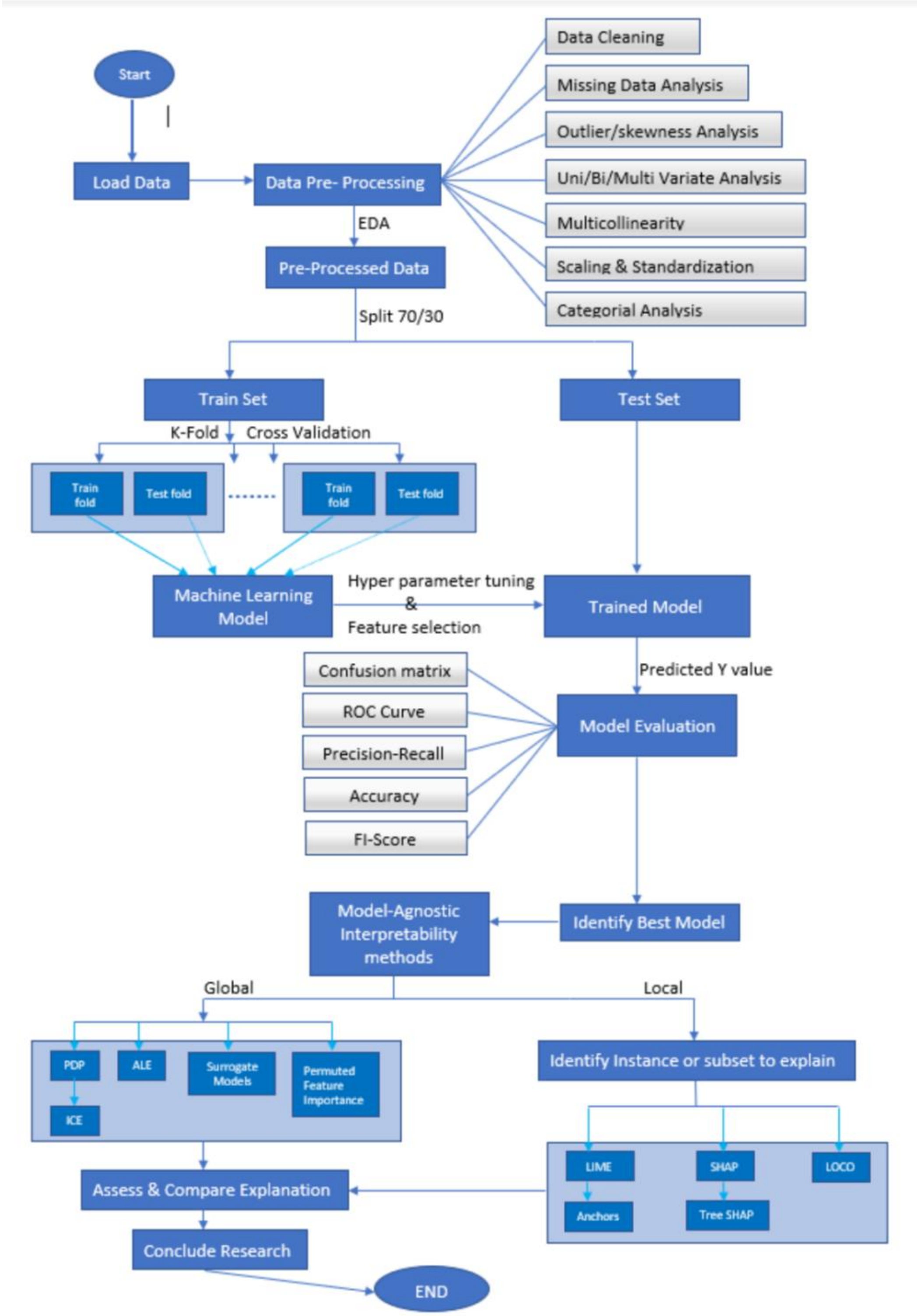
| | |
|---|---|
| *Assari, M. A., & Rahmati, M./ Driver drowsiness detection using face expression recognition/ IEEE International Conference on Signal and Image Processing Applications (ICSIPA)/2011* | The problems of sleepiness detecting systems were highlighted in this research study. Changes in intensity owing to lighting circumstances, the existence of spectacles, and the appearance of a beard on the person's face are all crucial considerations. Paper propose and construct a hardware solution based on infrared light that may be utilised to solve these difficulties in this project. |
| *I. García, S. Bronte, L. M. Bergasa, J. Almazán, and J. Yebes/ Vision-based drowsiness detector for real driving conditions/ IEEE Intelligent Vehicles Symposium, Proceedings./2012* | This research paper divided overall system into three phases. Pre-processing is the initial step, which involves face and eye identification as well as normalisation. The second stage detects and characterises pupil position, combining it with adaptive lighting filtering to make the system capable of handling outside lighting situations. The last stage calculates PERCLOS using data from closed eyelids. An outside database was developed to test this technology, comprising of many trials conducted over the course of more than 25 driving hours. The findings of this simulation environment are published in an analysis about the performance of this proposal. |
| *Alshaqaqi, B., Baquhaizel, A. S., Amine Ouis, M. E., Boumehed, M., Ouamri, A., & Keche, M./ Driver drowsiness detection system/ 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA)/2013* | The Raspberry Pi, pi camera, OpenCV, and Matlab are used to create the driver drowsiness. By recording the picture of the driver, it detects blinking and determines the shutting and opening of the eyes. If the driver closes his or her eye for longer than expected, the system alters the driver by vibration. If the driver does not reply to the warning, a recorded call to the nearest hospital is issued. The pedestrian detection system employs ultrasonic sensors and a stereo vision model for the GSM model. It identifies and feels the impediments in front of the cars, ensuring the safety of both the pedestrian and the driver. |
| *Dwivedi, K., Biswaranjan, K., & Sethi, A./ Drowsy driver detection using representation learning/ IEEE International Advance Computing Conference (IACC)./2014* | This research provides an intelligent vision-based method for detecting driver sleepiness. Blink rate, eye closure, yawning, eye brow shape, and other hand-engineered facial traits have all been used in the past. The proposed technique employs convolutional neural network features to explicitly capture numerous latent face traits as well as complicated non-linear feature relationships. A softmax layer is used to determine if the driver is sleepy or not. As a result, this technology is utilised to notify drivers of tiredness or lack of concentration in order to avoid traffic accidents. |
| *RamalathaMarimuthu, A. Suresh, M. Alamelu and S.Kanagaraj/ Driver fatigue detection using image processing and accident prevention, / International journal of pure and applied mathematics, vol. 116./2017* | In this study, a mechanism is developed to detect if the driver is drowsy by detecting the driver's eye gaze while driving the automobile. Image processing is used for analysis and detection, and a hardware-based warning system to inform the driver and others, as well as a control system to stop the car after determining its location and that of neighbouring cars, is built. |

| | |
|---|---|
| *Subbarao, A., Sahithya, K./ Driver Drowsiness Detection System for Vehicle Safety, and Exploring Engineering/ International Journal of Innovative Technology (IJITEE)/2019* | The goal of this work is to build an automated system to protect drivers from dangerous driving. The technology is set up in such a way that it can examine the eye blink in great detail. The eye blink of the driver is detected in this article utilising an IR-based eye blink sensor. As each eye blinks, the disparity across the eye will change. If the eye is closed, the output is high; otherwise, the output is low. It denotes whether an eye is closed or open. The circuit receives the IR output to indicate the alert. |
| *Sukrit Mehta, Sharad Dadhich, Sahil Gumber, Arpita Jadhav Bhatt/ Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio / International Conference on Sustainable Computing in Science, Technology and Management./2019* | The majority of classic sleepiness detection methods are based on behavioural factors, while some are obtrusive and may distract drivers, and others need costly sensors. As a result, in this research work, a simple, real-time sleepiness detection system for drivers is created and deployed on an Android application. Using image processing techniques, the system captures the videos and recognises the driver's face in each frame. |
| *Tayab Khan, M., Anwar, H., Ullah, F., Ur Rehman, A., Ullah, R., Iqbal, A., Kwak, K. S./ Smart Real-Time Video Surveillance Platform for Drowsiness Detection Based on Eyelid Closure. / Wireless Communications and Mobile Computing/2019* | The research paper determines tiredness by assessing whether a person's eyes are open or closed. The face of the individual is detected in the image as a first stage. To identify the curvature of the eyelids, the eyes are targeted and filtered with an extended Sobel operator in the identified face. Concavity is utilised to determine whether the eyelids are closed or open once the curves have been discovered. As a result, a concave upward curve indicates that the eyelid is closed, whereas a concave downward curve indicates that the eyelid is open. The suggested approach is also built on hardware so that it may be utilised in real-time circumstances like detecting driver fatigue. |

## III. RESEARCH METHODOLOGY

The Research for detection of Driver Drowsiness through different machine learning algorithm involves key processes such as selection of dataset, preprocessing the selected data, Feature extraction, Feature selection, comparison of different machine learning algorithms and Classification of Driver Alertness as 1. Open /Alert 2. Close/ Drowsiness Following are the high-level steps in which Research will be carried out [8][9]

- Driver Drowsiness Dataset available on Kaggle.com consist of pictures into two classes 1. Open/Alert 2. Close/Drowsiness, the data is already divided into two-part first part will be utilized for training Purpose and the second one for testing purpose. Feature selection will be done by various classification algorithm.
- Preprocessing of selected dataset -Not Required as dataset is clean and can directly utilize for modeling purpose.
- Selecting the appropriate features and extracting it from data by different algorithm.
- Feature selection is done from the extracted ones and different feature selection techniques are performed
- Results of different machine learning algorithms are compared with each other on train and test data set and then the performance will be compared on the basis of accuracy.




Research Methodology – Flowchart

### 3.1 DATA SET DESCRIPTION

The dataset used for this model is available on Kaggle website. We separated them into their respective labels 'Open' or 'Close'. The data was manually cleaned by removing the unwanted images which were not necessary for building the model. The data comprises around 2467 images for training purpose and 432 images for validation purpose. After training the model on our dataset, we compare the different models, and the best model will be used for detecting the Driver Drowsiness. We can use this model to classify if a person's eye is open/Alert or close/Drowsiness [2].

### 3.2 DRIVER DROWSINESS DETECTION SYSTEM

In this Python project, OpenCV will be used for gathering the images from webcam and feed them into selected model which will classify whether the person's eyes are 'Open/Alert' or 'Close/Drowsiness'. The approach for this Python project is as follows [3][4]:

- Take image as input from a camera.
- Detect the face in the image and create a Region of Interest (ROI).
- Detect the eyes from ROI and feed it to the classifier.
- Classifier will categorize whether eyes are open or closed.
- Choose the best model based on evaluation matrix
- Calculate score to check whether the person is drowsy.

### 3.3 MODEL BUILDING

As first step, the dataset will be split in Train – Test – Validation

- In real world scenario, dataset is usually divided with 70:30 ratio where 70% data will be considered as training data and 30% will be treated as test dataset (hidden).
- As data points is already available on Kaggle site in which total 2467 images which will be used for training purpose and 432 images is used for testing purpose. Total training data is categorized into 1. Open(alert) and 2. Close(drowsiness)

### 3.4 MODEL EVALUATION

Model selection can't be based on accuracy of the model. There are other metrics such as precision, recall, confusion matrix, F1 score for classification problems and can be chosen wisely based on requirement [5][6][7].

- The proportion of correctly classified observations is measured by accuracy which we are going to use for model evaluation in this Research Paper.
- Confusion matrix is representation of a 2x2 table of actual vs predicted class in binary classification which includes four parameters true positives, false positives, true negatives, and false negatives.
- Precision- Recall and Sensitivity-Specificity are two widely used evaluation metrics based on business requirement. Here sensitivity = recall.
- Best threshold would be one at which the True Positive Rate is high and False Positive Rate is low, i.e., misclassifications are low.
- Depending upon the problem statement, study will identify which measure is more important: high precision or high recall.
- F1-Score is the harmonic mean of precision and recall values for a classification problem where the requirement is of having best precision and recall at the same time. The formula for F1-Score is: 2*(precision*recall)/ (precision+ recall) [10][14]

## IV. ANALYSIS OF MODELS

Measures such as Accuracy, Precision-Recall are available for classification problems and can be chosen judiciously as required. Accuracy- Recall and sensitivity-Specificity are two commonly used evaluation parameters based on business needs. We have applied all seven Classification algorithm and Calculate confusion matrix, precision, sensitivity, f1-score and accuracy. The Model will use Accuracy Parameter for Comparison purpose.

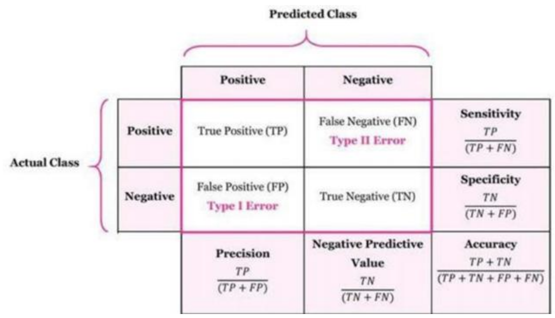
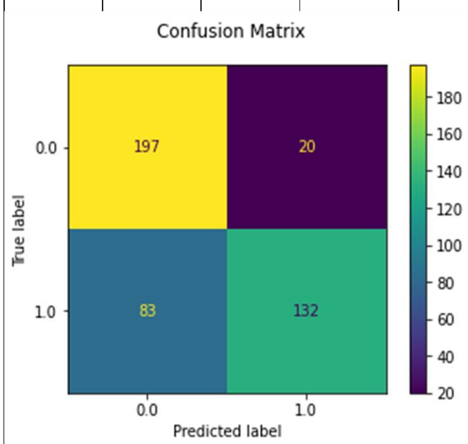


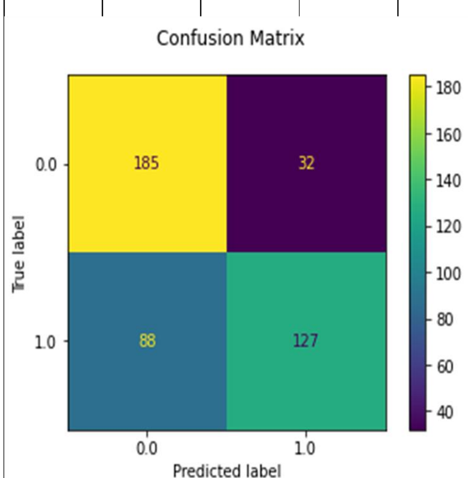

Figure 2: Confusion matrix

**Support Vector Machine**



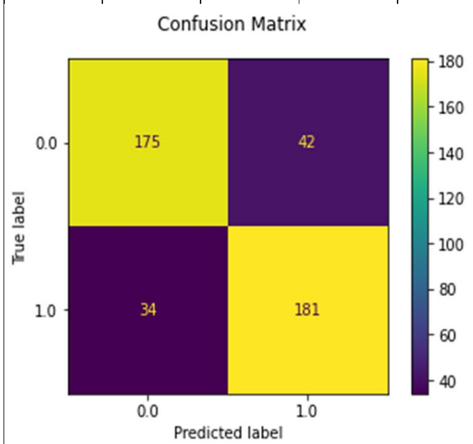

| | | | | |
|---|---|---|---|---|
| Classification report for classifier SVC(gamma=0.001): | | | | |
| | precision | recall | f1-score | support |
| 0 | 0.7 | 0.91 | 0.79 | 217 |
| 1 | 0.87 | 0.61 | 0.72 | 215 |
| Accuracy | | | 0.76 | 432 |
| Macro Avg | 0.79 | 0.76 | 0.76 | 432 |
| Weighted Avg | 0.79 | 0.76 | 0.76 | 432 |
| Accuracy of the Algorithm: | | | | 0.762 |

**Gaussian Naive Bayes**



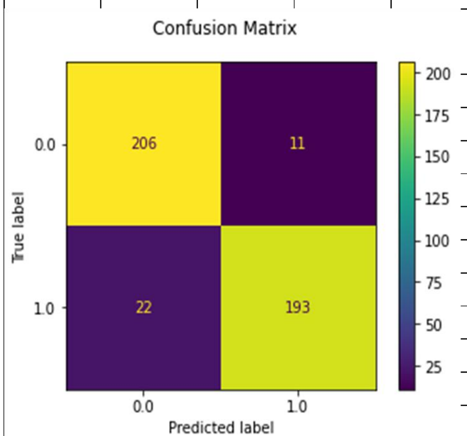

| | | | | |
|---|---|---|---|---|
| Classification report for classifier GaussianNB(): | | | | |
| | precision | recall | f1-score | support |
| 0 | 0.68 | 0.85 | 0.76 | 217 |
| 1 | 0.8 | 0.59 | 0.68 | 215 |
| accuracy | | | 0.72 | 432 |
| macro avg | 0.74 | 0.72 | 0.72 | 432 |
| weighted avg | 0.74 | 0.72 | 0.72 | 432 |
| Accuracy of the Algorithm: | | | | 0.722 |

**Decesion Tree**



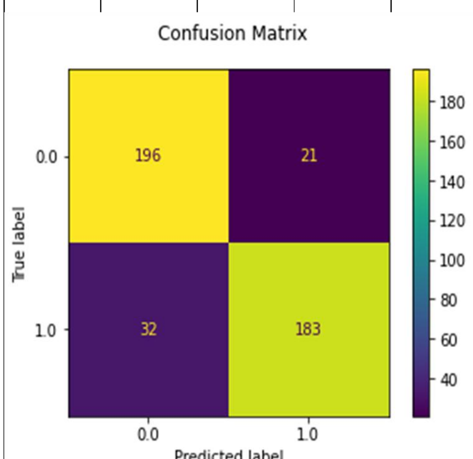

| Classification report for classifier DecisionTreeClassifier(): | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| 0 | 0.84 | 0.81 | 0.82 | 217 |
| 1 | 0.81 | 0.84 | 0.83 | 215 |
| accuracy | | | 0.82 | 432 |
| macro avg | 0.82 | 0.82 | 0.82 | 432 |
| weighted avg | 0.82 | 0.82 | 0.82 | 432 |
| Accuracy of the Algorithm: | | | | 0.824 |

Random Forest



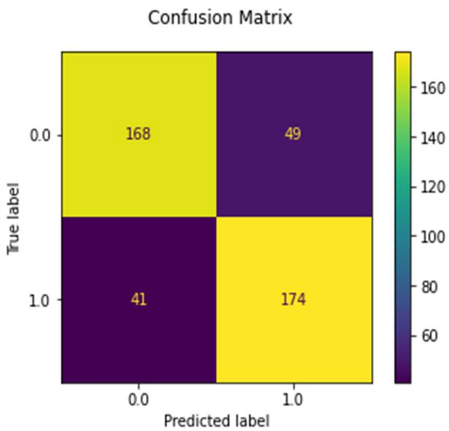

| Classification report for classifier RandomForestClassifier(max_depth=15, random_state=0): | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| 0 | 0.9 | 0.95 | 0.93 | 217 |
| 1 | 0.95 | 0.9 | 0.92 | 215 |
| accuracy | | | 0.92 | 432 |
| macro avg | 0.92 | 0.92 | 0.92 | 432 |
| weighted avg | 0.92 | 0.92 | 0.92 | 432 |
| Accuracy of the Algorithm: | | | | 0.924 |

**K Nearest Neighbors**




| Classification report for classifier KNeighborsClassifier(metric='euclidean', n_neighbors=2): | | | | |
|---|---|---|---|---|
| | precision | recall | f1-score | support |
| 0 | 0.86 | 0.9 | 0.88 | 217 |
| 1 | 0.9 | 0.85 | 0.87 | 215 |
| accuracy | | | 0.88 | 432 |
| macro avg | 0.88 | 0.88 | 0.88 | 432 |
| weighted avg | 0.88 | 0.88 | 0.88 | 432 |
| Accuracy of the Algorithm: | | | | 0.877 |

**Stochastic Gradient Descent**

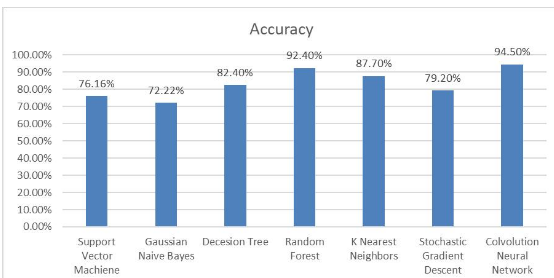



| | | | | |
|---|---|---|---|---|
| Classification report for classifier SGDClassifier(max_iter=8): | | | | |
| | precision | recall | f1-score | support |
| | | | | |
| 0 | 0.8 | 0.77 | 0.79 | 217 |
| 1 | 0.78 | 0.81 | 0.79 | 215 |
| | | | | |
| accuracy | | | 0.79 | 432 |
| macro avg | 0.79 | 0.79 | 0.79 | 432 |
| weighted avg | 0.79 | 0.79 | 0.79 | 432 |
| | | | | |
| Accuracy of the Algorithm: | | | | 0.792 |

| Convolution Neural Network(RELU and SOFTMAX Activation Function) |
|---|
| Epoch 1/15 |
| 77/77 [==============================] - 12s 152ms/step - loss: 0.5353 - accuracy: 0.7109 - val_loss: 0.4439 - val_accuracy: 0.7837 |
| Epoch 2/15 |
| 77/77 [==============================] - 12s 152ms/step - loss: 0.4139 - accuracy: 0.8000 - val_loss: 0.2563 - val_accuracy: 0.8100 |
| Epoch 3/15 |
| 77/77 [==============================] - 12s 151ms/step - loss: 0.3255 - accuracy: 0.8493 - val_loss: 0.2729 - val_accuracy: 0.8700 |
| Epoch 4/15 |
| 77/77 [==============================] - 11s 147ms/step - loss: 0.2769 - accuracy: 0.8727 - val_loss: 0.2723 - val_accuracy: 0.8750 |
| Epoch 5/15 |
| 77/77 [==============================] - 12s 150ms/step - loss: 0.2391 - accuracy: 0.9006 - val_loss: 0.4803 - val_accuracy: 0.8575 |
| Epoch 6/15 |
| 77/77 [==============================] - 12s 153ms/step - loss: 0.1904 - accuracy: 0.9179 - val_loss: 0.4785 - val_accuracy: 0.8900 |
| Epoch 7/15 |
| 77/77 [==============================] - 11s 149ms/step - loss: 0.1387 - accuracy: 0.9462 - val_loss: 0.3014 - val_accuracy: 0.9400 |
| Epoch 8/15 |
| 77/77 [==============================] - 12s 150ms/step - loss: 0.1317 - accuracy: 0.9470 - val_loss: 0.0632 - val_accuracy: 0.9200 |
| Epoch 9/15 |
| 77/77 [==============================] - 11s 148ms/step - loss: 0.0882 - accuracy: 0.9671 - val_loss: 0.1770 - val_accuracy: 0.9375 |
| Epoch 10/15 |
| 77/77 [==============================] - 11s 147ms/step - loss: 0.0612 - accuracy: 0.9782 - val_loss: 0.1424 - val_accuracy: 0.9450 |
| Epoch 11/15 |
| 77/77 [==============================] - 11s 146ms/step - loss: 0.0608 - accuracy: 0.9766 - val_loss: 0.2347 - val_accuracy: 0.9300 |
| Epoch 12/15 |
| 77/77 [==============================] - 12s 152ms/step - loss: 0.0530 - accuracy: 0.9819 - val_loss: 0.1430 - val_accuracy: 0.9450 |
| Epoch 13/15 |
| 77/77 [==============================] - 12s 156ms/step - loss: 0.0314 - accuracy: 0.9894 - val_loss: 0.2622 - val_accuracy: 0.9325 |
| Epoch 14/15 |
| 77/77 [==============================] - 11s 145ms/step - loss: 0.0344 - accuracy: 0.9881 - val_loss: 0.2270 - val_accuracy: 0.9400 |
| Epoch 15/15 |
| 77/77 [==============================] - 11s 147ms/step - loss: 0.0331 - accuracy: 0.9879 - val_loss: 0.0933 - val_accuracy: 0.9279 |
| **Colvolution Neural Network(Sigmoid and SOFTMAX Activation Function)** |
| Epoch 1/15 |
| 77/77 [==============================] - 12s 151ms/step - loss: 0.6236 - accuracy: 0.6559 - val_loss: 0.4650 - val_accuracy: 0.7380 |
| Epoch 2/15 |
| 77/77 [==============================] - 12s 159ms/step - loss: 0.4613 - accuracy: 0.7606 - val_loss: 0.3604 - val_accuracy: 0.7750 |
| Epoch 3/15 |
| 77/77 [==============================] - 12s 156ms/step - loss: 0.3915 - accuracy: 0.7926 - val_loss: 0.3205 - val_accuracy: 0.8025 |
| Epoch 4/15 |
| 77/77 [==============================] - 12s 152ms/step - loss: 0.3355 - accuracy: 0.8156 - val_loss: 0.2966 - val_accuracy: 0.8200 |
| Epoch 5/15 |
| 77/77 [==============================] - 12s 155ms/step - loss: 0.3085 - accuracy: 0.8370 - val_loss: 0.2573 - val_accuracy: 0.8500 |
| Epoch 6/15 |
| 77/77 [==============================] - 12s 153ms/step - loss: 0.3110 - accuracy: 0.8480 - val_loss: 0.3027 - val_accuracy: 0.8225 |
| Epoch 7/15 |
| 77/77 [==============================] - 12s 151ms/step - loss: 0.2629 - accuracy: 0.8739 - val_loss: 0.2117 - val_accuracy: 0.8525 |
| Epoch 8/15 |
| 77/77 [==============================] - 12s 153ms/step - loss: 0.2562 - accuracy: 0.8752 - val_loss: 0.5095 - val_accuracy: 0.8625 |
| Epoch 9/15 |
| 77/77 [==============================] - 11s 149ms/step - loss: 0.2193 - accuracy: 0.8940 - val_loss: 0.2995 - val_accuracy: 0.8250 |
| Epoch 10/15 |
| 77/77 [==============================] - 11s 149ms/step - loss: 0.2049 - accuracy: 0.9075 - val_loss: 0.2523 - val_accuracy: 0.9000 |
| Epoch 11/15 |
| 77/77 [==============================] - 11s 147ms/step - loss: 0.1548 - accuracy: 0.9335 - val_loss: 0.2757 - val_accuracy: 0.9025 |
| Epoch 12/15 |
| 77/77 [==============================] - 11s 149ms/step - loss: 0.1394 - accuracy: 0.9425 - val_loss: 0.1354 - val_accuracy: 0.9050 |
| Epoch 13/15 |
| 77/77 [==============================] - 12s 150ms/step - loss: 0.1217 - accuracy: 0.9520 - val_loss: 0.1307 - val_accuracy: 0.8975 |
| Epoch 14/15 |
| 77/77 [==============================] - 11s 145ms/step - loss: 0.1021 - accuracy: 0.9614 - val_loss: 0.2638 - val_accuracy: 0.8975 |
| Epoch 15/15 |
| 77/77 [==============================] - 11s 147ms/step - loss: 0.1063 - accuracy: 0.9569 - val_loss: 0.1860 - val_accuracy: 0.9159 |

Summary:

| Algorithm | Accuracy |
|---|---|
| Support Vector Machine | 76.16% |
| Gaussian Naive Bayes | 72.22% |
| Decision Tree | 82.40% |
| Random Forest | 92.40% |
| K Nearest Neighbours | 87.70% |
| Stochastic Gradient Descent | 79.20% |
| Convolution Neural Network | 94.50% |




## V. CONCLUSION

As per the Analysis result Convolution Neural Network is the best performing Algorithm in Driver Drowsiness Data Set and its accuracy (max) is 94.5% in respective of activation function which will be used in algorithm. Research paper here present the result of CNN when at inner side RELU Activation function and at end SOFTMAX Function has been used and CNN when at inner side SIGMOID Activation function and at end SOFTMAX Function has been used. We find that the Max Accuracy is 94.5% in case of RELU-SOFTMAX combination.

The second-best algorithm is random forest whose accuracy is 92.4%. KNN perform better than Decision Tree which is very interesting phenomena. The performance of SGD is poorest in all binary classification algorithm with 72.22% accuracy.

Over all if anybody want to make Driver Drowsiness Alarm based system, they must used CNN based model in order to find out better accuracy.

## VI. REFERENCES

[1] Association for Safe International Road Travel (ASIRT), Road Crash Statistics.http://asirt.org/initiatives/informingroadusers/road-safety-facts/road-crash-statistics, 2016

[2] https://data-flair.training/blogs/download-driver-drowsiness-detection-project-data/

[3] Journal of VLSI Signal Processing 23, 497–511 (1999) c °1999 Kluer Academic Publishers.Manufactured in The Netherlands.

[4] https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_de tection.html

[5] Eye Detection Using Morphological and Color Image Procesing Tanmay Rajpathaka, Ratnesh Kumar and Eric Schwartzb

[6] A Robust Algorithm for Eye Deteuction on Grey Intensity Face without Spectacles- JCS&T Vol. 5 No. 3

[7] Froba Kebbuck: Audio- and Video-Based Biometric Person Authentication, 3rd International Conference, AVBPA 2001, Halmstad, Sweden, June 2001. Proceedings,Springer. ISBN 3-540-42216-1.

[8] Flores, M.J., Armingol, J.M. & de la Escalera, A. Real-Time Warning System for Driver Drowsiness Detection Using Visual Information. *J Intell Robot Syst* **59,** 103–125 (2010). https://doi.org/10.1007/s10846-009-9391-1

[9] A. Eskandarian and A. Mortazavi, "Evaluation of a Smart Algorithm for Commercial Vehicle Driver Drowsiness Detection," 2007 IEEE Intelligent Vehicles Symposium, 2007, pp. 553-559, doi: 10.1109/IVS.2007.4290173.

[10] A. Deshwal and S. K. Sharma, "Twitter sentiment analysis using various classification algorithms," 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2016, pp. 251-257, doi: 10.1109/ICRITO.2016.7784960.

[11] https://analyticsindiamag.com/7-types-classification-algorithms/

[12] https://www.projectpro.io/article/7-types-of-

classification-algorithms-in-machine-learning/435

[13] Aher, Sunita B., and L. M. R. J. Lobo. "Comparative study of classification algorithms." *International Journal of Information Technology* 5.2 (2012): 239-243.

[14] Liu, Weihuang, et al. "Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection." *Future Internet* 11.5 (2019): 115.