iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72279

Predicting Employee Under Stress for Preemptive Remediation Using Machine Learning Algorithms

K.Tulasi Krishna Kumar[1], N.Dharmaraju Reddy[2]
[1]Assistant Professor & Training & Placement Officer, [2]MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract: The modern world is filled with stress. A person's pressure is affected by a variety of factors. Representatives in IT are more likely to be under pressure due to work pressure, overburdening, higher worker mastery, and so on. When a person is stressed, it can lead to a variety of mental health issues such as depression, anxiety, somatization, lack of concentration, andsoon.Asaresult,itisnecessarytoidentifyhumanstressatan earlystageinordertoprovide appropriate solutions and alleviate stress. There has been a lot of research done on stress prediction. Many research papers use Machine Learning techniques to predict stress, and many papersuse IOT -based sensors to extract the features needed for stress prediction. There are so many existing systems in predicting employee stress. In this project we are going to predict the employee stress using XGBOOST algorithm since it gives more Accuracy. Based on this the prediction we will remediate the persons under stress in the early stage which is good for their health and as well as the work.

Index Terms: Stress Detection, Employee Mental Health, Machine Learning, XGBoost Algorithm, Preemptive Remediation, Workplace Analytics, Predictive Modeling, Human Resources, Early Detection, Classification Metrics.

I. INTRODUCTİON

OnMarch11,2020,theWorldHealthOrganization(WHO)reportedcoronavirus(COVID-19)a pandemic that signifies a global, epidemic disorder frightening the entire universe [3]. COVID- 19 is a contagious disease affected by the corona virus. 'Corona viruses' are a huge family of virusesthatcauseailmentsvaryingfromthetypicalflutoothercriticalcomplications.According to WHO, on March 31, 2020, thevirus had reached 202 countries. Dueto this, stockmarkets and othersectorshaveexperiencedaseveredownturn ingrowth.This,inturn,affectsemployeestoo, who feel stressed when they are unable to cope with prolonged uncertainty and pressure. The application of machine learning and artificial intelligence to the field of business is seeing a lotof promising growth. The pattern of employee behavior is analyzed in [11].Vis-à-vis, they do not have any satisfaction due to long working hours in addition to having a heavy workload. Here, the foremost objective of this research is to analyze the consequence of stress on employee appearance. Moreover, this influences physical ailments and a lack of commitment to work. However, in the contemporary situation, COVID-19 has put the world populationinanunprecedentedposition.Through thiswork,weintendtoanalyzethestresslevel that employees are subjected to owing to a phenomenon like the present pandemic. Here,machine learning algorithms are used to predict whether employees undergoing stress or not.

II. LITERATURE SURVEY

Employee stress has become a critical concern in organizational performance and well-being. Traditional stress assessment methods—such as surveys, interviews, and self-reporting—are often subjective, infrequent, and prone to bias. Recent advancements in machine learning (ML) have enabled researchers to develop data-driven models that can analyze behavioral, physiological, and performance-related data to predict employee stress levels. For example, studies have utilized biometric signals (heart rate, skin conductance), workplace activity logs, email tone analysis, and wearable sensor data to detect early signs of stress. These systems aim to enable organizations to intervene proactively before stress negatively impacts health and productivity. Several machine learning techniques, including Logistic Regression, Support Vector Machines (SVM), Random Forests, and Deep Neural Networks, have been applied for stress prediction. Research by Sano and Picard (2013) showed that physiological data combined with ML can predict stress with high accuracy. Additionally, ensemble models and deep learning have been explored for improving predictive performance, particularly when dealing with high-dimensional and imbalanced datasets. Techniques like feature selection and time-series analysis help in refining model accuracy and responsiveness. These predictive models support timely and personalized interventions, making them valuable tools for mental health management in the workplace.

III. CHALLENGES

Data Collection Difficulties: Capturing accurate and relevant data (e.g., physiological signals, behavioral patterns, and productivity metrics) is challenging and often intrusive.

Privacy and Ethical Concerns: Monitoring employee behavior raises concerns about consent, data misuse, and workplace surveillance ethics.

Imbalanced Datasets: Stress-related instances are usually rare compared to normal conditions, causing models to become biased or less sensitive to stressed cases.

Individual Differences: Stress triggers vary greatly among individuals, making it hard to build generalized models that apply universally.

Feature Selection Complexity: Identifying the right features (such as email sentiment, biometric trends, or work patterns) that genuinely indicate stress requires domain expertise.

Real-time Prediction Demands: Implementing models that operate in real-time adds computational complexity and demands efficient, scalable systems.

Model Interpretability: Many advanced ML models (e.g., deep learning) function as black boxes, making it difficult for HR or management to understand or trust the results.

Labeling Data: Supervised learning requires labeled data, but accurately labeling stress levels is difficult due to its subjective nature.

Environmental and Cultural Factors: Organizational culture, work environment, and external factors can influence stress but are hard to quantify and incorporate into models.

Integration into Workflows: Seamlessly embedding stress detection models into existing HR systems or workplace tools without disrupting operations is often complex.

IV. PROPOSED METHODOLOGY

The proposed methodology involves a systematic approach using machine learning techniques to identify and manage employee stress proactively. The process begins with data collection from multiple sources such as wearable devices (heart rate, skin temperature), digital behavior (keyboard activity, mouse movement), workplace logs (attendance, workload), and communication data (email sentiment, chat frequency). All collected data is anonymized and processed to ensure privacy compliance.

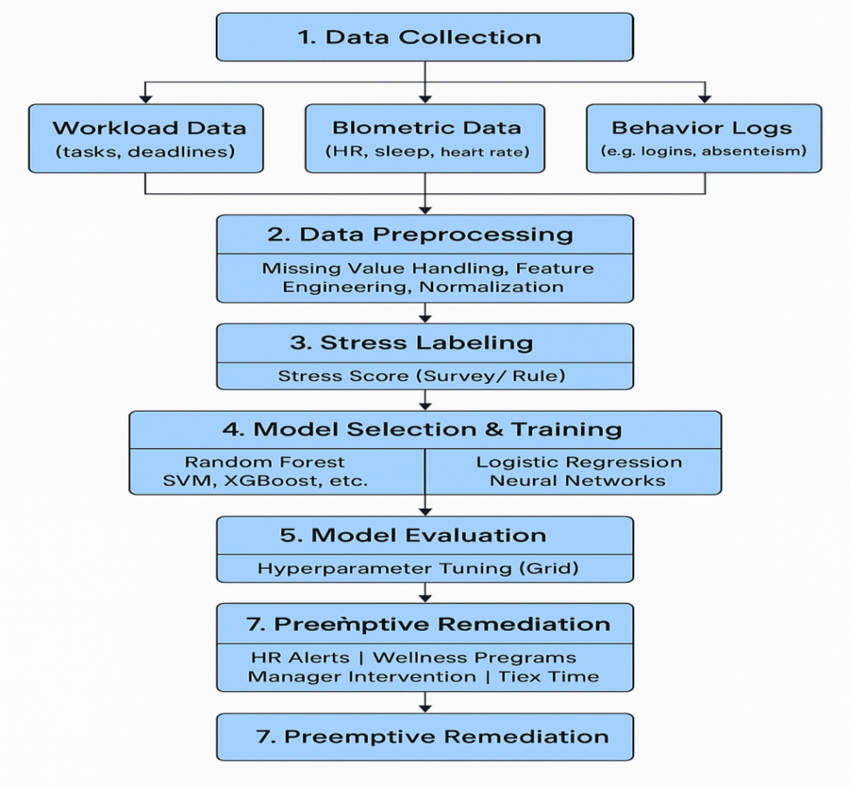

1. Data Collection
Workload Data
(tasks, deadlines)
Blometric Data
(HR, sleep, heart rate)
Behavior Logs
(e.g. logins, absenteism)
2. Data Preprocessing
Missing Value Handling, Feature Engineering, Normalization
3. Stress Labeling
Stress Score (Survey/ Rule)
4. Model Selection & Training
Random Forest
SVM, XGBoost, etc.
Logistic Regression
Neural Networks
5. Model Evaluation
Hyperparameter Tuning (Grid)
7. Preemptive Remediation
HR Alerts | Wellness Pregrams
Manager Intervention | Tiex Time
7. Preemptive Remediation

Figure 1: Flow chart of the proposed methodology

ADVANTAGES:
The proposed methodology offers several key advantages over existing systems for stress prediction in the workplace:
High Accuracy and Performance: The use of the XGBoost algorithm, known for its efficiency and accuracy, ensures that the stress prediction model provides more reliable results compared to traditional classifiers like SVM or Naive Bayes.
Early Stress Detection: By analyzing employee-related data, the system can detect stress at an early stage, enabling preemptive interventions before the stress escalates into serious mental or physical health issues.
No External Hardware Required: Unlike IoT-based or biometric systems, this approach does not rely on sensors or wearable devices, making it more cost-effective and easier to implement across organizations.
Handles Mixed Data Types: The model is capable of working with datasets containing both numerical and categorical features, using label encoding and normalization for pre-processing.
Feature Importance Analysis: XGBoost provides built-in feature importance metrics, which help HR teams understand which factors (like department, workload, gender, etc.) are most indicative of employee stress.

V. MACHINE LEARNING ALGORITHMS

A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithmwill produce an estimate about a pattern in the data. An Error Function: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model. A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met.

A. TypesofMachineLearningMethods

1) Supervisedmachinelearning

Supervised learning also known as supervised machine learning, is defined by its use of labelled datasetstotrain algorithmsthatto classifydataorpredictoutcomesaccurately.Asinputdatais fed into the model, it adjusts its weights until the model has been fitted appropriately. Thisoccursaspartofthecrossvalidationprocesstoensurethatthemodelavoidsoverfittingorunder fitting. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve Bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and more.

2) Unsupervisedmachinelearning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyses and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its abilityto discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross selling strategies, customer segmentation, image and pattern recognition [6]. It's also used to reduce the number of features in a model through the process of dimensionality reduction; principal component analysis (PCA) and singular value decomposition (SVD) are two commonapproaches for this. Other algorithms used in unsupervised learning include neural networks, kmeans clustering, probabilistic clustering methods, and more [7].

3) Semi-supervisedlearning

Semi-supervised learning offers a happymedium between supervised and unsupervised learning. During training, it uses a smaller labelled data set to guide classification and feature extraction from a larger, unlabelled data set [8]. Semi-supervised learning can solve the problem of having not enough labelled data (or not being able to afford to label enough data) to train a supervised learning algorithm.

4) PracticalUseofMachineLearning

Speech Recognition: It is also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, and it is a capability which uses natural language processing (NLP) to process human speech into a written format. Many mobile devices incorporate speech recognition into their systems to conduct voice search—e.g. Siri—or provide more accessibility around texting.

5) *Customer Service:*

Online chat bots are replacing human agents along the customer journey. They answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms [10]. Examples include messaging bots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger, and tasks usually done by virtual assistants and voice assistants.

6) *Computer Vision:*

This AI technology enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks [11]. Powered by convolutional neural networks, computer vision has applications within photo tagging in social media, radiology imaging in healthcare, and self driving cars within the automotive industry.

7) *Recommendation Engines:*

Using past consumption behavior data, AI algorithms can help to discover data trends that can be used to develop more effective cross-selling strategies. This is used to make relevant add-on recommendations to customers during the checkout process for online retailers [12].

Automated stock trading: Designed to optimize stock portfolios, AI-driven high-frequency tradingplatformsmakethousand sorevenmillionsoftradesperdaywithout human intervention.

VI. ARCHITECTURE

The proposed system employs a comprehensive set of techniques involving data preprocessing, transformation, and supervised learning to accurately predict employee stress. Initially, the dataset undergoes data preprocessing, where missing values are handled by replacing them with zero to ensure consistency. Additionally, irrelevant columns such as Employee ID, which do not contribute to the prediction process, are dropped from the dataset to reduce noise and dimensionality. NeXT, the system uses Label Encoding to convert categorical variables (e.g., Gender, Department) into numeric values. This transformation is crucial because machine learning algorithms require numerical inputs. The Label Encoder class from the Scikit-learn library is applied for this purpose. To standardize the range of feature values and ensure uniform influence across attributes, Min-Max Normalization is applied. This technique scales each feature to a range between 0 and 1, improving the efficiency and convergence speed of the model during training. After preprocessing, the dataset is split into two subsets—90% for training and 10% for testing. This Train-Test Split approach helps evaluate the model's generalization ability on unseen data. The primary learning algorithm used is XGBoost, a powerful and optimized implementation of gradient boosting. XGBoost operates by building an ensemble of decision trees, where each new tree is designed to correct the errors made by the previous trees, leading to high-performance classification. Finally, the model's performance is evaluated using a variety of evaluation techniques, including Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC Curve.

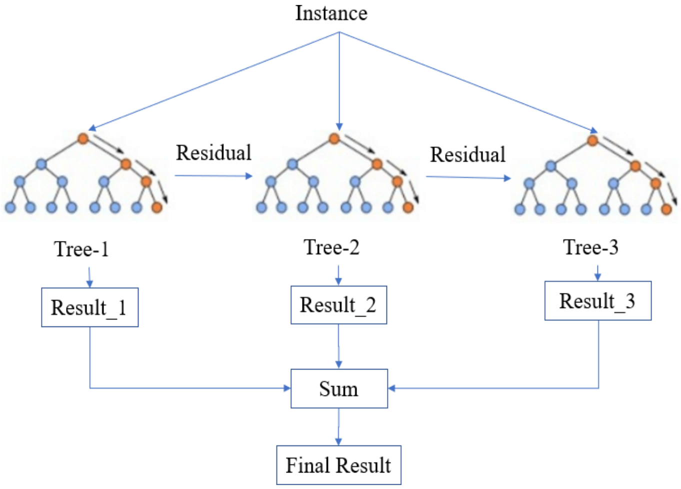


Figure: Architecture of Proposed system

A. TOOLS:

To implement and evaluate the proposed system for predicting employee stress using machine learning, several development tools, programming libraries, and platforms are utilized. These tools facilitate data handling, preprocessing, model training, evaluation, and visualization.

1) Programming Language:

Python is chosen for its rich ecosystem of libraries, simplicity, and efficiency in handling data science and machine learning tasks.

2) Development Environment:

Jupyter Notebook / Google Colab / Anaconda these platforms are used for coding, visualizing results, and step-by-step development. They support inline display of charts and interactive data exploration.

3) Libraries and Frameworks:

Library	Purpose
Pandas	Data manipulation and analysis (loading and processing CSV files)
NumPy	Numerical operations and array handling
Matplotlib	Visualization of data and model performance
Sea born	Advanced visualization (e.g., heatmaps, bar plots)
Scikit-learn	Machine learning preprocessing tools like LabelEncoder, MinMaxScaler, train-test split, and evaluation metrics
XGBoost	The core machine learning algorithm used for classification
OS	File handling and path operations

B. METHODS:

The proposed system for predicting employee stress employs a supervised machine learning approach that integrates various data processing and classification methods to ensure accurate and meaningful predictions. The entire methodology is implemented using Python, with essential libraries such as Scikit-learn, XGBoost, Pandas, and Matplotlib. The process begins with data collection, where a structured dataset containing employee attributes—such as gender, department, and job role—is used. The collected dataset may contain both numeric and categorical variables, as well as missing values, which are addressed in the data preprocessing stage. Missing values are filled (e.g., with zero), and irrelevant fields like Employee ID are removed to streamline the dataset. Next, Label Encoding is applied to convert all categorical (non-numeric) features into numeric format so that machine learning algorithms can process them. Following this, Min-Max Normalization is performed to bring all feature values within a common scale, improving model convergence and ensuring uniform feature contribution. The cleaned dataset is then split into training and testing sets, typically with a 90:10 ratio. This is essential for evaluating the model's performance on unseen data. The primary method for classification is the XGBoost algorithm, which builds an ensemble of decision trees using gradient boosting techniques. It is selected for its high accuracy, ability to handle complex data, and built-in regularization to prevent overfitting.After training the model, it is used to predict stress levels in the testing data, classifying employees as either stressed (1) or non-stressed (0). Finally, the system employs multiple evaluation techniques—such as Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC Curve—to assess the effectiveness and reliability of the model. This combination of preprocessing, transformation, training, and evaluation forms the core methodology of the system, enabling early detection of employee stress for timely and targeted intervention.

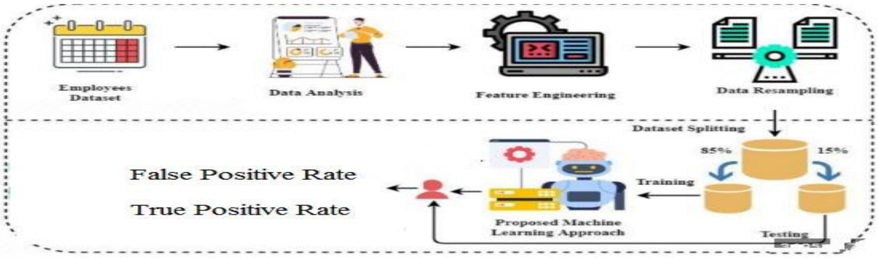
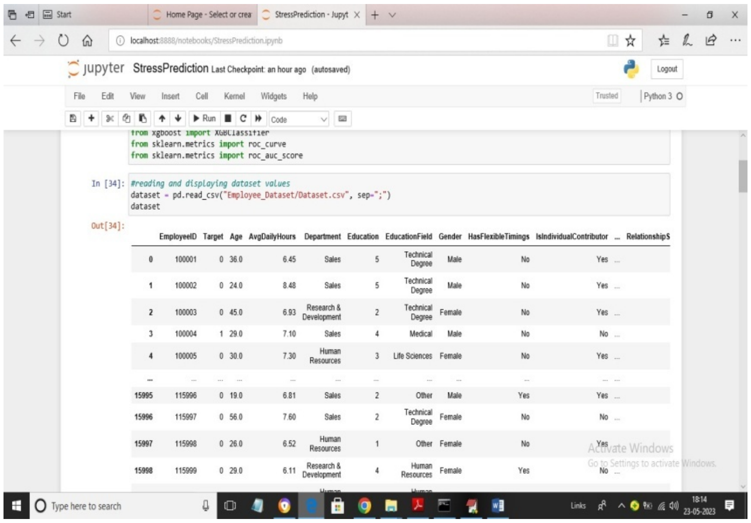


Figure 3: System architecture

C. TESTING

Software testing is a critical process for identifying faults—programming errors that may lead to failures under specific conditions—and ensuring overall software quality. It involves multiple stages, including unit testing to validate individual modules using white-box techniques, integration testing to verify interactions between components, system testing to confirm that the complete system meets requirements, and acceptance testing to demonstrate functionality to the client. Two key testing approaches are black-box testing, which assesses system behavior based on inputs and outputs without considering internal code, and white-box testing, which focuses on internal logic and execution paths. A structured test plan outlines the process, resources, and schedule, with test cases comparing expected and actual results. The testing process concludes with reports that document outcomes and guide

D. OUTPUTS:


Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:14
23-05-2023

```
from xgboost import XGBClassifier
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

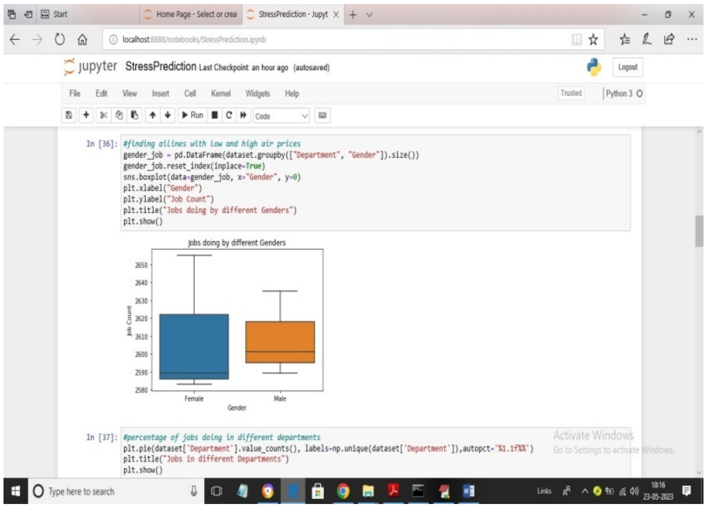
```
In [34]: #reading and displaying dataset values
         dataset = pd.read_csv("Employee_Dataset/Dataset.csv", sep=";")
         dataset
```

Out[34]:

	EmployeeID	Target	Age	AvgDailyHours	Department	Education	EducationField	Gender	HasFlexibleTimings	IsIndividualContributor	...	RelationshipS
0	100001	0	36.0	6.45	Sales	5	Technical Degree	Male	No	Yes	...	
1	100002	0	24.0	8.48	Sales	5	Technical Degree	Male	No	Yes	...	
2	100003	0	45.0	6.93	Research & Development	2	Technical Degree	Female	No	Yes	...	
3	100004	1	29.0	7.10	Sales	4	Medical	Male	No	No	...	
4	100005	0	30.0	7.30	Human Resources	3	Life Sciences	Female	No	Yes	...	
...	...	...	...	...	...	...	...	...	...	...	...	
15995	115996	0	19.0	6.81	Sales	2	Other	Male	Yes	Yes	...	
15996	115997	0	56.0	7.60	Sales	2	Technical Degree	Female	No	No	...	
15997	115998	0	26.0	6.52	Human Resources	1	Other	Female	No	Yes	...	
15998	115999	0	29.0	6.11	Research & Development	4	Human Resources	Female	Yes	No	...	

Screenshot 2: Displaying the Datasets

In above screen reading and displaying dataset values and in above dataset values some are numeric and some are non-numeric but XGBOOST accept only non-numeric data so by employing label encoder class we can convert all non-numeric data into numeric values. Label encoder class assigned unique integer ID to each non-numeric value such as Gender MALE will get 0 and FEMALE will get 1.

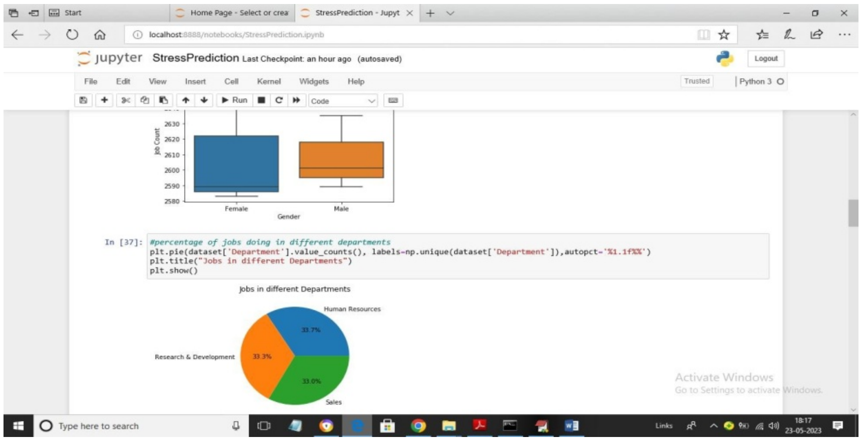

Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Jobs doing by different Genders
2650
2640
2630
2620
2610
2600
2590
2580
Job Count
Female
Male
Gender
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:16
23-05-2023

```
In [36]: #finding ailines with low and high air prices
         gender_job = pd.DataFrame(dataset.groupby(["Department", "Gender"]).size())
         gender_job.reset_index(inplace=True)
         sns.boxplot(data=gender_job, x="Gender", y=0)
         plt.xlabel("Gender")
         plt.ylabel("Job Count")
         plt.title("Jobs doing by different Genders")
         plt.show()
```

```
In [37]: #percentage of jobs doing in different departments
         plt.pie(dataset['Department'].value_counts(), labels=np.unique(dataset['Department']),autopct='%1.1f%%')
         plt.title("Jobs in different Departments")
         plt.show()
```

Screenshot 3: Displaying the Floating graphs

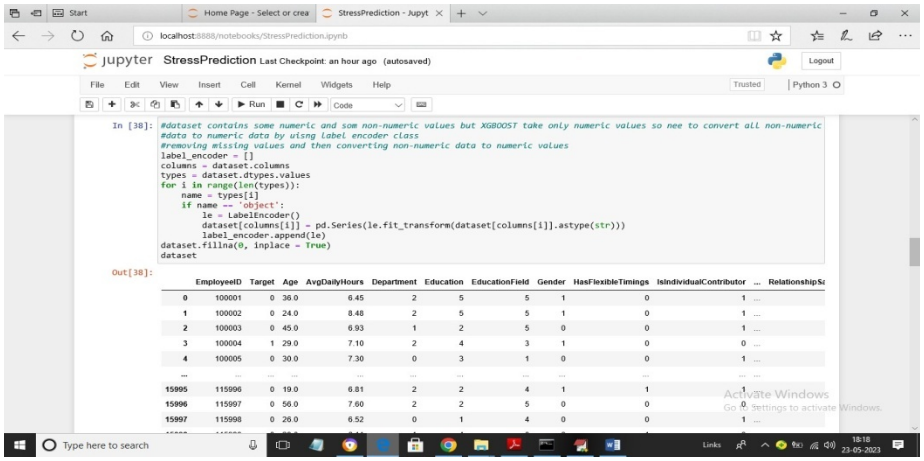
Inabovegraphwearefindingplottinggraphofgenderdoingjobswherex-axisrepresents gender and y-axis represents count


Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Job Count
2630
2620
2610
2600
2590
2580
Female
Male
Gender
Jobs in different Departments
Human Resources
33.7%
Research & Development
33.3%
33.0%
Sales
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:17
23-05-2023

```
In [37]: #percentage of jobs doing in different departments
         plt.pie(dataset['Department'].value_counts(), labels=np.unique(dataset['Department']),autopct='%1.1f%%')
         plt.title("Jobs in different Departments")
         plt.show()
```

Screenshot 4: Finding the Floating graphs

Inabovegraphwearefindingandplottinggraphofnumberofjobsavailableindifferent departments


Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:18
23-05-2023

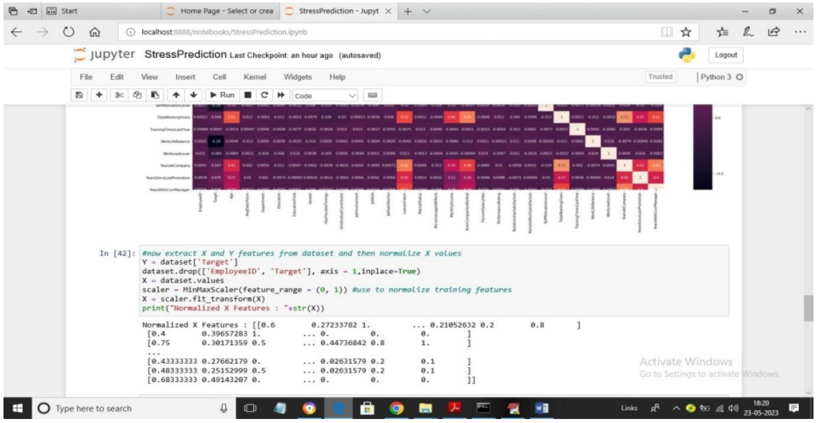
```
In [38]: #dataset contains some numeric and som non-numeric values but XGBOOST take only numeric values so nee to convert all non-numeric
         #data to numeric data by uisng label encoder class
         #removing missing values and then converting non-numeric data to numeric values
         label_encoder = []
         columns = dataset.columns
         types = dataset.dtypes.values
         for i in range(len(types)):
             name = types[i]
             if name == 'object':
                 le = LabelEncoder()
                 dataset[columns[i]] = pd.Series(le.fit_transform(dataset[columns[i]].astype(str)))
                 label_encoder.append(le)
         dataset.fillna(0, inplace = True)
         dataset
```

Out[38]:

	EmployeeID	Target	Age	AvgDailyHours	Department	Education	EducationField	Gender	HasFlexibleTimings	IsIndividualContributor	...	RelationshipSa
0	100001	0	36.0	6.45	2	5	5	1	0	1	...	
1	100002	0	24.0	8.48	2	5	5	1	0	1	...	
2	100003	0	45.0	6.93	1	2	5	0	0	1	...	
3	100004	1	29.0	7.10	2	4	3	1	0	0	...	
4	100005	0	30.0	7.30	0	3	1	0	0	1	...	
...	...	...	...	...	...	...	...	...	...	...	...	
15995	115996	0	19.0	6.81	2	2	4	1	1	1	...	
15996	115997	0	56.0	7.60	2	2	5	0	0	0	...	
15997	115998	0	26.0	6.52	0	1	4	0	0	1	...	

Screenshot 5: Conversion of non-numeric to numeric values

In above screen we are converting all non-numeric data to numeric values and after conversionwe can see all values are numeric only

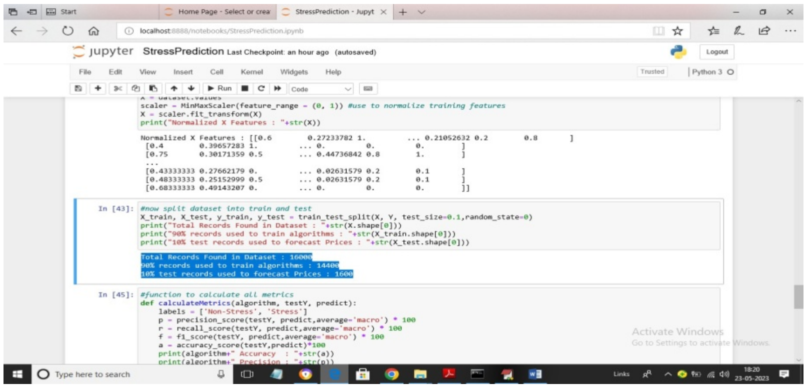

Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
WorkLifeBalance
WorkLoadLevel
YearsAtCompany
YearsSinceLastPromotion
YearsWithCurrManager
-0.2
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:20
23-05-2023

```
In [42]: #now extract X and Y features from dataset and then normalize X values
         Y = dataset['Target']
         dataset.drop(['EmployeeID', 'Target'], axis = 1,inplace=True)
         X = dataset.values
         scaler = MinMaxScaler(feature_range = (0, 1)) #use to normalize training features
         X = scaler.fit_transform(X)
         print("Normalized X Features : "+str(X))

         Normalized X Features : [[0.6        0.27233782 1.         ... 0.21052632 0.2        0.8        ]
          [0.4        0.39657283 1.         ... 0.         0.         0.        ]
          [0.75       0.30171359 0.5        ... 0.44736842 0.8        1.        ]
          ...
          [0.43333333 0.27662179 0.         ... 0.02631579 0.2        0.1        ]
          [0.48333333 0.25152999 0.5        ... 0.02631579 0.2        0.1        ]
          [0.68333333 0.49143207 0.         ... 0.         0.         0.        ]]
```

Screen shot 6: Out puts

Inabovescreenweare extractingXandYfeaturesandthennormalizingallX trainingvalues


Start
Home Page - Select or crea
StressPrediction - Jupyt
localhost:8888/notebooks/StressPrediction.ipynb
jupyter StressPrediction Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
18:20
23-05-2023

```
X = dataset.values
scaler = MinMaxScaler(feature_range = (0, 1)) #use to normalize training features
X = scaler.fit_transform(X)
print("Normalized X Features : "+str(X))

Normalized X Features : [[0.6        0.27233782 1.         ... 0.21052632 0.2        0.8        ]
 [0.4        0.39657283 1.         ... 0.         0.         0.         ]
 [0.75       0.30171359 0.5        ... 0.44736842 0.8        1.         ]
 ...
 [0.43333333 0.27662179 0.         ... 0.02631579 0.2        0.1        ]
 [0.48333333 0.25152999 0.5        ... 0.02631579 0.2        0.1        ]
 [0.68333333 0.49143207 0.         ... 0.         0.         0.         ]]

In [43]: #now split dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.1,random_state=0)
print("Total Records Found in Dataset : "+str(X.shape[0]))
print("90% records used to train algorithms : "+str(X_train.shape[0]))
print("10% test records used to forecast Prices : "+str(X_test.shape[0]))

Total Records Found in Dataset : 16000
90% records used to train algorithms : 14400
10% test records used to forecast Prices : 1600

In [45]: #function to calculate all metrics
def calculateMetrics(algorithm, testY, predict):
    labels = ['Non-Stress', 'Stress']
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
    print(algorithm+" Accuracy  : "+str(a))
    print(algorithm+" Precision : "+str(p))
```

Screen shot 7: Out puts

In above screen we are splitting dataset into train and test where application using 80% datasetfor training and 20% for testing

VII.CONCLUSION

To evaluate our model to achieve a better performance which is done by using XGB classifier. This is one of the best optimization technique and this is like a decision tree-based algorithm which adopts gradient boosting frame work technique for analysis and confusion matrix which tells us how many correct values are predicted by our model. XG Boost has tremendous predictive power and is about 10 times more durable than other gradient boosting techniques. It holds a varietyof regularization which diminishes overfitting and enhances overall performance. Consequently, it is further recognized as the “regularized boosting” technique. Like it has true positive, true negative, false positive, false negative values. Used to evaluate the performance of the classification model

VIII. ACKNOWLEDGEMENTS

Kandhati Tulasi Krishna Kumar: Training & Placement Officer with 16 years’ experience in training & placing the students into IT, ITES & Core profiles & trained more than 9,700 UG, PG candidates & trained more than 450 faculty through FDPs. Authored various books for the benefit of the diploma, pharmacy, engineering & pure science graduating students. He is a Certified Campus Recruitment Trainer from JNTUA, did his Master of Technology degree in CSE from VTA and in process of his Doctoral research. He is a professional in Pro-E, CNC certified by CITD He is recognized as an editorial member of IJIT (International Journal for Information Technology & member in IAAC, IEEE, MISTE, IAENG, ISOC, ISQEM, and SDIWC. He published 6 books, 55 articles in various international journals on Databases, Software Engineering, Human Resource Management and Campus Recruitment & Training.

Ruttala Kondalarao is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning r kondalarao has taken up her PG project on predicting employee under stress for preemptive remedation Using Machine Learningand published the paper in connection to the project under the guidance of K.Tulasi Krishna Kumar SVPEC.

REFERENCES

[1] Shekhar Pandey, Supriya Muthuraman, Abhilash Shrivastava.The International Symposium on Intelligent Systems Technologies and Applications (2018), DOI: 10.1007/978-3-319- 68385- 0_10.

[2] Ramachandran, R; Rajeev, D.C; Krishnan, S.G; Subathra.P. International Journal of Applied Engineering Research (2015), Research India Publications, Volume 10,Number 10, p.25433- 25448

[3] RaminZibaseresht: How to Respond to the Ongoing Pandemic Outbreak of the Coronavirus Disease (COVID-19) (WHO- World Health Organization) (2020), ISSN 2349- 8870.

[4] Chen, Tianqi; Guestrin, Carlos; “XG Boost: A scalable Tree Boosting System”. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and DataMining, San Francisco, USA (2016). ACM. pp. 785-794.

[5] Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002,XXIX,487p.28 illus. ISBN 978-0-387-95442-4.

[6] Manuela Aparicio and Carlos J. Costa “Data Visualization”. Communication Design Quarterly (2014). DOI: 10.1145/2721882.2721883.

[7] Ali.M., Alqahtani.A.,Jones.M.W., Xie.X .“Clustering and classification for Time Series Data in Visual Analytics: A Survey IEEE Access 7,8930535, pp. 181314-181338.

[8] Mouubayedd.A, Injadat.M.,Nassif, Lutfuyya, H.Shami, A E-learning: Challenges and Research Opportunities Using Machine Learning and Data Analytics (2018) IEEE Access 6,8417405. pp. 39117-39138.

[9] Kim., Soyata, T., Behnagh, R.F. Towards Emotionally Aware AI Smart Classroom: Current Issues and Directions for Engineering and Education (2018) IEEE pp. 5308-5331.

[10] PriyonesiS.Madeh; EI-Diraby Tamer E. “Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems”. Journal of Transprotation Engineering, Pavements (2020). DOI: 10.1061/JPEODX.0000175.

[11] AbhijeetRawal, SnehaMhatre.IOSRJournal ofBussiness and Management (IOSR-JBM), e-ISSN:2278-487X,P-ISSN:2319

[12] JanneSkakon, Karina Nielsen, Vihelm Borg, Jaime Gazman. An international Journal of work, Health and organizations, Volume 24,2010-Issue 2.

[13] K. S. Santosh and S. H. Bharathi, "Non-negative matrix factorization algorithms for blind source sepertion in speech recognition," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017,pp.2242-2246,doi: 10.1109/RTEICT.2017.8256999.

[14] Rajendra Prasad P, N. Narayan, S. Gayathri and S. Ganna, “An Efficient E-Health Monitoring with Smart Dispensing System for Remote Areas”, 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2018, pp. 2120-2124. doi: 10.1109/RTEICT42901.2018.9012480