Detection of Stress in It Employees Using Machine Learning Technique

ABSTRACT

This project describes the Mental Health Prediction using Data Mining in medicine is an emerging field of great importance to provide a prognosis and deeper understanding of disease classification, specifically in Mental Health areas. This project describes the mental stress condition of IT employs using machine learning algorithm. Mental stress condition is detected using data analysis for an employee. Stress in IT employees is a common problem that can lead to negative outcomes such as burnout, decreased job satisfaction, and increased turnover rates. Identifying factors that contribute to stress in IT employees can help organizations implement interventions to prevent or mitigate stress. This study aims to investigate the relationship between various factors and stress in IT employees using logistic regression analysis. The objective of this paper is to apply machine learning and visual processing to identify overworked IT employees. Our technology is an improved version of older stress detection systems that did not include live detection or personal counselling. Stress detection methods that don't include real-time monitoring or individual counselling are being updated in this research. A survey is used to collect data on employees' mental stress levels in order to provide effective stress management solutions. In order to get the most out of your employees, this paper will look at stress management and how to create a healthy, spontaneous work environment.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Mental health includes our emotional, psychological, and social well-being. It affects how we think, feel, and act. Mental wellness of an individual is the state of mind of that person and also provides an indication of his/her general nature. Mental illness is an outcome of imbalances in brain chemistry. The evaluation of mental wellness is extremely critical to understand and suggest therapies for patients with a deviated mental behavior. There is a fundamental change to incorporate the mental fitness outline of an affected individual by healthcare providers and it will be made obligatory in the approaching years to deliver improved medication and also promote speedy recoveries.

MACHINE LEARNING

Predictive analytics tools are powered by several different models and algorithms that can be applied to wide range of use cases. Determining what predictive modelling techniques are best for your company is key to getting the most out of a **predictive** analytics solution and leveraging data to make insightful decisions. In the statistical context, Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalization in order to get a system that performs well on yet unseen data instances Machine learning is a relatively new discipline within Computer Science that provides a collection of data analysis techniques. Some of these techniques are based on well-established statistical methods (e.g. logistic regression and principal component analysis) while many others are not.

Most statistical techniques follow the paradigm of determining a particular probabilistic model that best describes observed data among a class of related models. Similarly, most machine learning techniques are designed to find models that best fit data (i.e. they solve certain optimization problems), except that these machine learning models are no longer restricted to probabilistic ones.

Therefore, an advantage of machine learning techniques over statistical ones is that the latter require underlying probabilistic models while the former do not. Even though some machine learning techniques use probabilistic models, the classical statistical techniques are most often too stringent for the oncoming Big Data era, because data sources are increasingly complex and multi-faceted. Prescribing probabilistic models relating variables from disparate data sources that are plausible and amenable to statistical analysis might be extremely difficult if not impossible.

Machine learning might be able to provide a broader class of more flexible alternative analysis methods better suited to modern sources of data. It is imperative for statistical agencies to explore the possible use of machine learning techniques to determine whether their future needs might be better met with such techniques than with traditional ones.

1.3.1 CLASSES OF MACHINE LEARNING

1.3.1.1 Examples of supervised learning

Logistic regression, when used for prediction purposes, is an example of supervised machine learning. In logistic regression, the values of a binary response variable (with values 0 or 1, say) as well as a number of predictor variables (covariates) are observed for a number of observation units. These are called training data in machine learning terminology. The main hypotheses are that the response variable follows a Bernoulli distribution (a class of probabilistic models), and the link between the response and predictor variables is the relation

that the logarithm of the posterior odds of the response is a linear function of the predictors. The response variables of the units are assumed to be independent of each other, and the method of maximum likelihood is applied to their joint probability distribution to find the optimal values for the coefficients (these parameterize the aforementioned joint distribution) in this linear function. The particular model with these optimal coefficient values is called the "fitted model," and can be used to "predict" the value of the response variable for a new unit (or, "classify" the new unit as 0 or 1) for which only the predictor values are known. Support Vector Machines (SVM) are an example of a non-statistical supervised machine learning technique; it has the same goal as the logistic regression classifier just described: Given training data, find the best-fitting SVM model, and then use the fitted SVM model to classify

1.3.1.2 Machine learning

Machine learning (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

The most widely used predictive models are

- **Decision trees:** Decision trees are a simple, but powerful form of multiple variable analysis. They are produced by algorithms that identify various ways of splitting data into branch-like segments. Decision trees partition data into subsets based on categories of input variables, helping you to understand someone's path of decisions.

- **Regression (linearandlogistic)**: Regression is one of the most popular methods in statistics. Regression analysis estimates relationships among variables, finding key patterns in large and diverse data sets and how they relate to each other.
- **Neural networks**: Patterned after the operation of neurons in the human brain, neural networks (also called artificial neural networks) are a variety of deep learning technologies. They're typically used to solve complex pattern recognition problems – and are incredibly useful for analyzing large data sets. They are great at handling nonlinear relationships in data – and work well when certain variables are unknown

Developing the right environment

While machine learning and predictive analytics can be a boon for any organization, implementing these solutions haphazardly, without considering how they will fit into everyday operations, will drastically hinder their ability to deliver the insights the organization needs

Understanding predictive models

Typically, an organization's data scientists and IT experts are tasked with the development of choosing the right predictive models – or building their own to meet the organization's needs. Today, however, predictive analytics and machine learning is no longer just the domain of mathematicians, statisticians and data scientists, but also that of business analysts and consultants. More and more of a business' employees are using it to develop insights and improve business operations – but problems arise when employees do not know what model to use, how to deploy it, or need information right away.

At SAS, we develop sophisticated software to support organizations with their data governance and analytics. Our data governance solutions help

organizations to maintain high-quality data, as well as align operations across the business and pinpoint data problems within the same environment. Our predictive analytics solutions help organizations to turn their data into timely insights for better, faster decision making. These predictive analytics solutions are designed to meet the needs of all types of users and enables them to deploy predictive models rapidly.

OBJECTIVE

- The objective of the project is detection of mental stress condition using machine learning condition.

- The data analysis from the ID employees based on their work condition and data's are used for predict the stress condition.

PROBLEM STATEMENT

- The problem of detecting stress in IT employees using Machine Learning (ML) can be addressed using a supervised ML approach.

- This approach involves training a model using labeled data that contains information on employees' stress levels.

- The data could include items such as job satisfaction, working hours, salary, and job security.

- The model would then be used to predict the stress levels of new employees based on their characteristics.

- In order to effectively use ML for this purpose, it is important to ensure that the model is able to accurately identify the different types of stress. Additionally, the model should be able to identify the underlying causes of employee stress.

- This will enable the model to provide more accurate predictions and insights into the best ways to address stress in IT employees.

CHAPTER-2

LITERATURE SURVEY

2.1 TITLE: Educational Data Mining: Classification Techniques for Recruitment Analysis

AUTHOR: Siddu P. Algur Prashant Bhat Nitin Kulkarni

YEAR: 2019

DESCRIPTION: Data Mining is a dominant tool for academic and educational field. Mining data in education atmosphere is called Educational Data Mining. Educational Data Mining is concerned with developing new methods to discover knowledge from educational/academic database and can be used for decision making in educational/academic systems. This work demonstrates an effective mining of students performance data in accordance with placement/recruitment process. The mining result predicts weather a student will be recruited or not based on academic and other performance during the entire course. To mine the students' performance data, the data mining classification techniques such as – Decision tree Random Tree and J48 classification models were built with 10 cross validation fold using WEKA. The constructed classification models are tested for predicting class label for new instances. The performance of the classification models used are tested and compared. Also the misclassification rates for the classification experiment are analyzed. Educational data mining is becoming an emerging trend nowadays. In this work, under the educational data mining theme we made an effective attempt to predict recruitment of students based on their academic and other performances. This helps students as well as educational institutions to know about students, those be recruited by the industry before starting of the campus recruitment process.

2.2 TITLE: Predicting IT Employability Using Data Mining Techniques

AUTHOR: Keno C. Piad Melvin A. Ballera Menchita Dumlao Shaneth C. Ambat

YEAR: 2020

DESCRIPTION: Researchers in higher education are beginning to explore the potential of data mining in analyzing data for the purpose of giving quality service and needs of their graduates. Thus, educational data mining emerges as one tools to study academic data to identify patterns and help for decision making affecting the education. This paper predicts the employability of IT graduates using nine variables. First, different classification algorithms in data mining were tested making logistic regression with accuracy of 78.4 is implemented. Based on logistic regression analysis, three academic variables directly affect; IT Core, IT Professional and Gender identified as significant predictors for employability. The data were collected based on the five year profiles of 515 students randomly selected at the placement office tracer study. The researchers compare five algorithms under classification technique on IT employability dataset. The results show that the Logistic regression, achieved the highest accuracy of 78.4%. The second highest accuracy was achieved using CHAID algorithm with an accuracy of 76.3 %. In addition, the study show that 3 possible predictors with a direct effect on IT employability are the IT Core, IT professionally and Gender. To determine the correctness of the system in predicting the employability and based on the existing job by the graduates, a CRT classification analysis was used and reveals that the system needs to have several trial data to increase its accuracy and reliability. The study can utilize newly acquired data to generate more rules and predict more accurately in IT employability.

2.3 TITLE: OER Recommendations to Support Career Development

AUTHOR: Mohammadreza Tavakoli Ali Faraji Stefan T. Mol

YEAR: 2020

DESCRIPTION: This Work in Progress Research paper departs from the recent, turbulent changes in global societies, forcing many citizens to re-skill themselves to (re)gain employment. Learners therefore need to be equipped with skills to be autonomous and strategic about their own skill development. Subsequently, high quality, on-line, personalized educational content and services are also essential to serve this high demand for learning content. Open Educational Resources (OERs) have high potential to contribute to the mitigation of these problems, as they are available in a wide range of learning and occupational contexts globally. However, their applicability has been limited, due to low metadata quality and complex quality control. These issues resulted in a lack of personalized OER functions, like recommendation and search. Therefore, we suggest a novel, personalized OER recommendation method to match skill development targets with open learning content. This is done by: 1) using an OER quality prediction model based on metadata, OER properties, and content; 2) supporting learners to set individual skill targets based on actual labour market information, and 3) building a personalized OER recommender to help learners to master their skill targets. Accordingly, we built a prototype focusing on Data Science related jobs, and evaluated this prototype with 23 data scientists in different expertise levels. Pilot participants used our prototype for at least 30 minutes and commented on each of the recommended OERs. As a result, more than 400 recommendations were generated and 80.9% of the recommendations were reported as useful.

2.4 TITLE: Mining Labor Market Requirements Using Distributional Semantic Models and Deep Learning

AUTHOR: Dmitriy Botov(B), Julius Klenin, Andrey Melnikov, Yuri Dmitrin, Ivan Nikolaev, and Mikhail Vinel

YEAR: 2019

DESCRIPTION: This article describes a new method for analyzing labor market requirements by matching job listings from online recruitment platforms with professional standards to weigh the importance of particular professional functions and requirements and enrich the general concepts of professional standards using real labor market requirements. Our approach aims to combat the gap between professional standards and reality of fast changing requirements in developing branches of economy. First, we determine professions for each job description, using the multi-label classifier based on convolutional neural networks. Secondly, we solve the task of concept matching between job descriptions and standards for the respective professions by applying distributional semantic models. In this task, the average word2vec model achieved the best performance among other vector space models. Finally, we experiment with expanding general vocabulary of professional standards with the most frequent unigrams and bigrams occurring in matching job descriptions. Performance evaluation is carried out on a representative corpus of job listings and professional standards in the field of IT. A method for mining the actual requirements of the labor market, based on a matching concepts between job listings and professional standards which allows us to deal with the gap, existing between some professional standards and ever-developing reality. In our first step, the multi-label classification of jobs by profession, the best results were achieved by a model based on a convolutional neural network trained on one-hot word embeddings of lemmatized documents.

2.5 TITLE: Developing Educational Programs Using Russian IT Job Market Analysis

AUTHOR: Andrey Sozykin Anton Koshelev Alexander Bersenev

YEAR: 2021

DESCRIPTION: The paper presents an approach to developing new educational programs for information technologies (IT) specialists using job market analysis. We developed a set of innovative software tools for collecting data from job sites and analyzing the data using machine learning algorithms. The tools were used to create a dataset of job advertisements from popular Russian job site HeadHunter for a period from 2006 until 2020. Both tools and dataset are freely available on the Internet. In order to develop the educational programs, we analyzed the demand for IT professions and programming languages, as well as joint occurrence of skills in job descriptions. To demonstrate the possibilities of our approach, two educational programs were created: a JavaScript-based program for Web Development and a Python-based program for Data Science. Developing the educational programs based on the requirements of the job market allows graduates to increase their employability. During the programs, students not only study the foundation of the profession such as programming languages, but also acquire a complete set of skills required to work in demanded positions. Developing the educational programs based on the requirements of the job market allows graduates to increase their employability. The programs are created for popular job positions such as Web Developer and Data Scientist. During the programs, students not only study the foundation of the profession, such as programming languages, but also acquire a complete set of skills required to work in demanded positions.

2.6 TITLE: Vocational Training Has an Influence on Employee Career Development: A Case Study Indonesia

AUTHOR: Indra Prasetyo, Universitas Wijaya Putra Indonesia Nabilah Aliyyah

YEAR: 2021

DESCRIPTION: This paper aims to test the effect employee job training has on employee career development in the company. This paper uses a quantitative approach with the process of finding the knowledge by using data in the form of numbers as a tool that can be generalized to prove hypotheses. The population in this study was 135 employees by sampling 100 employees using sampling method probability by using random sampling type samples to use slovin formula. Analysis techniques use multiple linear regressions. The results showed that job training influences the career development of employees in companies with a significant rate of 0.00 less than α=0.05. Previous research has been conducted to test job training on employee career development in large companies; the findings conclude that employee job training affects employee career development. This research researchers try to research medium-scale companies down. Based on the results of the hypothesis test, it explains that job training has an influence on development career company employees with more job training significance level smaller than 0.05, namely 0.000 <0.05. so that the t count is 24,583 and t table is 1,985. So, it can be concluded that t-count> t-table, then H0 is rejected and H1 is accepted. The magnitude of the influence of Job Training on Career Development can be seen in the R Square value of 86.2% and the remaining 13.8% is influenced by other variables outside of this study.

2.7 TITLE: Science, Technology and Innovation Policy Indicators and Comparisons of Countries through a Hybrid Model of Data Mining and MCDM Methods

AUTHOR: Gokhan Ozkaya , Mehpare Timor 3 and Ceren Erdin

YEAR: 2021

DESCRIPTION: Science, technology and innovation (STI) policies are of great importance for countries to reach their sustainable development goals. Numerous global databases have many indicators that measure and compare the performance of STI policies of countries. However, many problems arise regarding how to identify, classify and systematically analyze these indicators in order to measure, monitor and improve the performance of STI. The study includes a literature review on global problems and new trends in STI policies, while mentioning the necessity of an internationally comparable STI indicator set, current STI indicator studies and efforts, and studies for each continent. In light of these, all the indicators selected are introduced in detail. The strengths and weaknesses of the countries in the study in terms of evaluation indicator values are indicated. After determining the indicator weights objectively with the entropy method, 40 countries are compared with TOPSIS, VIKOR, PROMETHEE I-II, ARAS, COPRAS, MULTIMOORA, ELECTRE, SAW and MAUT methods. In addition, countries that show similarities with each other are evaluated by cluster analysis, which is one of the data mining classification methods. This study offers a new and original approach with MCDM methods on this subject. Considering all the results obtained in the study together, these rankings are compared among themselves and with the rankings specified in the Global Innovation (2019) and Global Competitiveness (2019) indices, and it is seen that the results are consistent. In addition, it is possible to update and publish this study every year with updated data.

2.8 TITLE: Effectiveness of corporate social responsibility (CSR) in implementation of social sustainability in warehousing of developing countries: A hybrid approach

AUTHOR: Sadia Samar Ali a, Rajbir Kaur

YEAR: 2021

DESCRIPTION: The field of sustainable supply chain management has been abundantly explored from environmental perspective, however the growing focus on holistic sustainable development demands critical perspective on social dimensions also. Consequently, organizations under national and global pressures tend to implement social sustainability measures in their supply chains. While existing research suggests that social sustainability measures are prevalent in developed countries, the developing nations nonetheless require elaborate efforts. This study aims to explore the initiatives and practices adopted by logistics, specifically warehousing operations organizations. Social sustainability practices identified and finalized using Best Worst Method (BWM) are ranked for their impact on improving social sustainability footprint of organizations. The effectiveness of corporate social responsibility (CSR) is evaluated based on the implementation of the social sustainability practices in the warehousing organizations of the developing countries. Taking a survey-based approach data collected from 217 organizations is statistically analyzed using binary logistic regression. The hybrid approach results indicate the importance of 'responsible work environment' as strong contender for reflecting effectiveness of corporate social responsibility. Contrary to theoretical predictions and expectations, 'training education and development' is found lacking for improving social footprints of organizations indicating deeper and rigorous approaches.

2.9 TITLE: Exploring the antecedents of institutional effectiveness: a case study of higher education universities in India

AUTHOR: Vidya Bai Gokarna, Suhan Mendon, Iqbal Thonse Hawaldar

YEAR: 2021

DESCRIPTION: The significance of Institutional Effectiveness is pivotal to the functioning of an academic institution. The mushrooming of private institutions in the Indian higher education space necessitates exploring its antecedents to ensure quality higher education is imparted by the institution. The purpose of this research endeavor is to explore the impact of Academic Leadership on Institutional Effectiveness with a mediating role of Campus Culture and Faculty Involvement in Decision Making. The study employed a cross-sectional research design and implemented a purposive sampling technique to collet primary data from 80 permanent faculties and 249 students of private engineering colleges located in the Karnataka state of India. Data was collected through a structured questionnaire and analyzed using Partial Least Square-Structural Equation Modeling. Hypothesis shows academic leadership and campus culture has high influence on institutional effectiveness. Faculty in decision making and campus culture partially mediates between academic leadership and institutional effectiveness. The results of FIMIX-PLS and PLS-MGA analysis shows the similarity in the results of total effect and path relationships. This paper provides theoretical foundations and empirical findings on conceptualizing the antecedents of institutional effectiveness. The outcomes of this research serve as significant input to policy makers and higher education institutions to facilitate enhancement of institutional effectiveness.

2.10 TITLE: Study and Prediction Analysis of the Employee Turnover using Machine Learning Approaches

AUTHOR: Raj Chakraborty; Krishna Mridha; Rabindra Nath Shaw; Ankush Ghosh

YEAR: 2021

DESCRIPTION: Employee Turnover is one of the key market challenges in Human Resource (HR) Analytics. Organizations usually invest a greater amount of money and time in the hiring of staff and nursing them in the hope to receive value addition. When an employee leaves the company, the reduction of opportunity costs is borne by the company. Turnover is especially prevalent in large-scale recruitment agencies. The risk of replacing workers remains important for most employers. This is due to the amount of time spent recruiting and selecting a successor, the sign-on incentives, and the lack of morale for several months as the new employee gets used to the new job. The tangible costs of workforce turnover will be the cost of recruiting new staff, the cost of recruitment and hiring, the time of transition, future product or service quality issues, the cost of temporary staff, the cost of training, the cost of lack of production, the cost of lost expertise and the cost of the job being empty before an acceptable replacement is found. We find that the attributes of workers such as Job Position, overtime, and work level affect significantly attrition. Various classification methods are introduced such as logistic regression, linear discriminate analysis, ridge classification, lasso classification, decision trees, and random forests to forecast and concurrently measure the likelihood of turnover of every new employee. Data from an HR department of the company available at Kaggle were used to estimate the employee turnover. The dataset includes 10 different attributes of 1470 personnel. Dataset specifies if the personnel is leaving or staying based on the attributes.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The prediction of mental health using algorithms are Decision Tree, SVM The technical and fundamental series analysis is used by the most of the predictions. Considers Stress, PTSD, and Traumaticbrain injuries related to sports. Mental health prediction is one of the most essential parts of reducing the probability of serious mental illness. Meanwhile, mental health prediction can provide a theoretical basis for public health department to work out psychological intervention plans for medical workers.

Stress is classified using supervised machine learning algorithms such as KNN classifiers in the proposed system. The detection of stress is accomplished through image processing. The worker's picture is given by the program as information, and Image Processing is utilized for discovery at the underlying stage. By translating an image to digital form and performing operations on it, image processing can improve or extract relevant information from it. In previous papers, various machine learning algorithms are used like SVM, linear regression, logistic regression, etc.., but didn’t use KNN for the experiment which is similar to our approach. Not only accuracy but we also found Classification Error, Sensitivity, specificity, false positive rate error, and precision. Our system is an updated version of prior stress detection systems that did not include live analysis or individual counselling, but it now incorporates live monitoring and frequent employee analysis, as well as identifying physically and emotionally stress levels. Because there is no continuous taking of photographs, it takes less time and produces more effective outcomes when compared with the results achieved by continuously capturing images of a person.

3.2 DISADVANTAGE

- Less accuracy.
- Low performance

3.3 PROPOSED SYSTEM

The proposed system stress condition of IT employs is predicted using deep learning algorithm. Deep learning XG Boost based deep learning model is deployed for detect the stress condition of IT Employees Here the project is split into training of datasets train the model and deploy model. Web UI is created predict the mental stress condition and give suggestion.

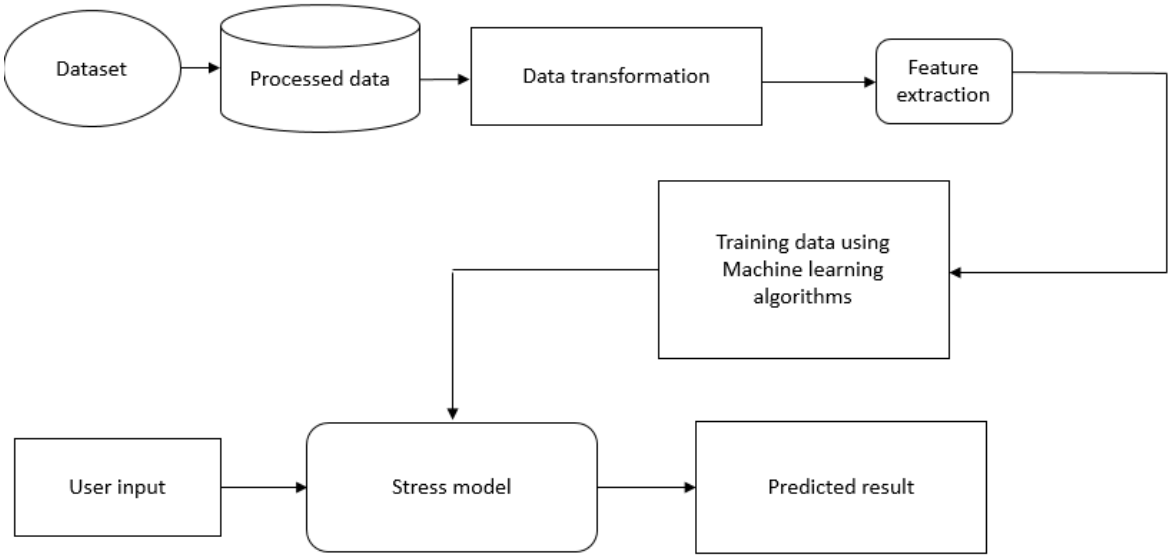
The proposed system would involve collecting data from IT employees using a structured questionnaire containing questions related to demographic factors and work-related factors. The data collected would be used to train an XGBoost model to predict stress levels in IT employees.

The XGBoost model would be trained using the collected data to identify the most important features that contribute to stress levels in IT employees. The model would then be used to predict stress levels for new employees based on their demographic and work-related factors. The system would provide insights into the factors that contribute to stress in IT employees and help organizations identify interventions to mitigate stress levels.

3.4 ADVANTAGE

- High accuracy
- Increase overall performance

3.5 SYSTEM ARCHITECTURE



3.6 ALGORITHM

XGBOOST

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for "Extreme Gradient Boosting" and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

XGBoost can be used in a variety of applications, including Kaggle competitions, recommendation systems, and click-through rate prediction, among others. It is also highly customizable and allows for fine-tuning of various model parameters to optimize performance.

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

What is XGBoost in Machine Learning?

XGBoost is designed for speed, ease of use, and performance on large datasets. It does not require optimization of the parameters or tuning, which means that it can be used immediately after installation without any further configuration.

XGBoost Features

XGBoost is a widespread implementation of gradient boosting. Let's discuss some features of XGBoost that make it so attractive.

- XGBoost offers regularization, which allows you to control overfitting by introducing L1/L2 penalties on the weights and biases of each tree. This feature is not available in many other implementations of gradient boosting.
- Another feature of XGBoost is its ability to handle sparse data sets using the weighted quantile sketch algorithm. This algorithm allows us to deal with non-zero entries in the feature matrix while retaining the same computational complexity as other algorithms like stochastic gradient descent.
- XGBoost also has a block structure for parallel learning. It makes it easy to scale up on multicore machines or clusters. It also uses cache awareness, which helps reduce memory usage when training models with large datasets.

- Finally, XGBoost offers out-of-core computing capabilities using disk-based data structures instead of in-memory ones during the computation phase.

XgBoost Formula

XgBoost is a gradient boosting algorithm for supervised learning. It's a highly efficient and scalable implementation of the boosting algorithm, with performance comparable to that of other state-of-the-art machine learning algorithms in most cases.

Why XGBoost?

XGBoost is used for these two reasons: execution speed and model performance.

Execution speed is crucial because it's essential to working with large datasets. When you use XGBoost, there are no restrictions on the size of your dataset, so you can work with datasets that are larger than what would be possible with other algorithms.

Model performance is also essential because it allows you to create models that can perform better than other models. XGBoost has been compared to different algorithms such as random forest (RF), gradient boosting machines (GBM), and gradient boosting decision trees (GBDT). These comparisons show that XGBoost outperforms these other algorithms in execution speed and model performance.

What Algorithm Does XGBoost Use?

Gradient boosting is a ML algorithm that creates a series of models and combines them to create an overall model that is more accurate than any individual model in the sequence.

It supports both regression and classification predictive modeling problems.

To add new models to an existing one, it uses a gradient descent algorithm called gradient boosting.

Gradient boosting is implemented by the XGBoost library, also known as multiple additive regression trees, stochastic gradient boosting, or gradient boosting machines.

XGBoost Benefits and Attributes

XGBoost is a highly portable library on OS X, Windows, and Linux platforms. It's also used in production by organizations across various verticals, including finance and retail.

XGBoost is open source, so it's free to use, and it has a large and growing community of data scientists actively contributing to its development. The library was built from the ground up to be efficient, flexible, and portable.

You can use XGBoost for classification, regression, ranking, and even user-defined prediction challenges! You can also use this library with other tools like H2O or Scikit-Learn if you want to get more out of your model-building process.

CHAPTER-4

IMPLEMENTATION

4.1 MODULES

- Data collection
- Data pre-processing
- EDA analysis
- Model development
- Machine learning based recommendation system
- Model deployment

4.2 MODULE DESCRIPTION

4.2.1 DATA COLLECTION

- The dataset for our project is collected form the website called Naukri.
- The dataset contains 17,458 records.
- The primary step in our data pipeline was to scrape the data from naukri.com.
- We used Scrapy for scraping the data, Selenium for automating the scraping process and stored the scraped data in the PostgreSQL database.
- From naukri.com we scraped job posts of data scientist, data engineer and data analyst and also various company reviews given by former and current employees.
- From naukri.com, we scraped job posts for the same titles and also interview questions that were asked by various companies for these titles.

4.2.2 DATA PREPROCESSING

- The NLP toolkit is used to extract the text from the data set.
- In this step, we preprocessed the data that we scraped in order to make it ready for data analysis. It involved tasks such as data cleaning and data integration.
- Since the description field was paragraph based, we removed all the stop words, punctuations to extract the keywords for NLP tasks.
- We integrated data because the job posts were scraped from two different websites and there were many duplicate listings as well.
- Hence we applied entity resolution technique, Jaccard similarity to identify the similar pairs after integration.

4.2.3 EDA ANALYSIS

- EDA is stands for Exploratory Data Analysis which is used to get insights of the dataset.
- The EDA analysis is represents the pictorial form the insights which helps to understand about the data.
- This stage involved finding the actual answers to the questions mentioned in the problem statement. For a job role, we find a similarity score of the job title + bigrams with the occupation titles in the O*Net database.
- We further calculate the similarity score of the job description with the corresponding O*Net occupation competencies.
- The top 20 competencies are returned for that job role. Also, we perform the same for company reviews to identify bad factors a company has and provide competencies that will improve their work values. The methodology section will give a deeper understanding of the analysis.

4.2.4 MODEL DEVELOPMENT

- Classification and clustering algorithms are used in the process. Based on the better result, the model will be selected.
- We integrated all the exciting insights that we got from the job market world in our interactive dashboard which has all the findings and results that any user can understand effectively.
- We displayed the results in the form of bar charts, line charts, tables, and word clouds. These visualizations are simple and meaningful.

4.2.5 RECOMMENDATION SYSTEM

- The selected best algorithm has been used for the recommendation process.
- The recommendation system will provide the skill set and needed course to get the jobs in the IT companies.

4.2.6 MODEL DEPLOYMENT

- Django is used to build the web application in the module.

CHAPTER 5

SYSTEM SPECIFICATION

H/W SYSTEM CONFIGURATION:-

- processor - Pentium – IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

S/W SYSTEM CONFIGURATION:-

- Operating System : Windows 7 or 8
- Software : python Idle

SOFTWARE ENVIRONMENT

Python Technology:

Python is an interpreter, high-level, general-purpose programming language. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. **Python** is often described as a "batteries included" language due to its comprehensive standard library.

Python Programing Language:

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by Meta programming and met objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

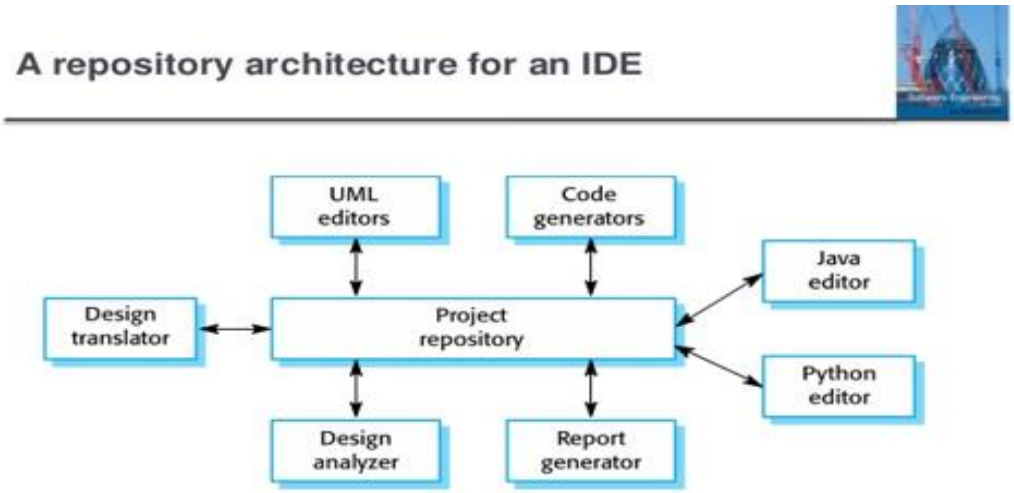
Python packages with a wide range of functionality, including:

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Cross-platform Language
- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library
- GUI Programming Support
- Integrated

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.



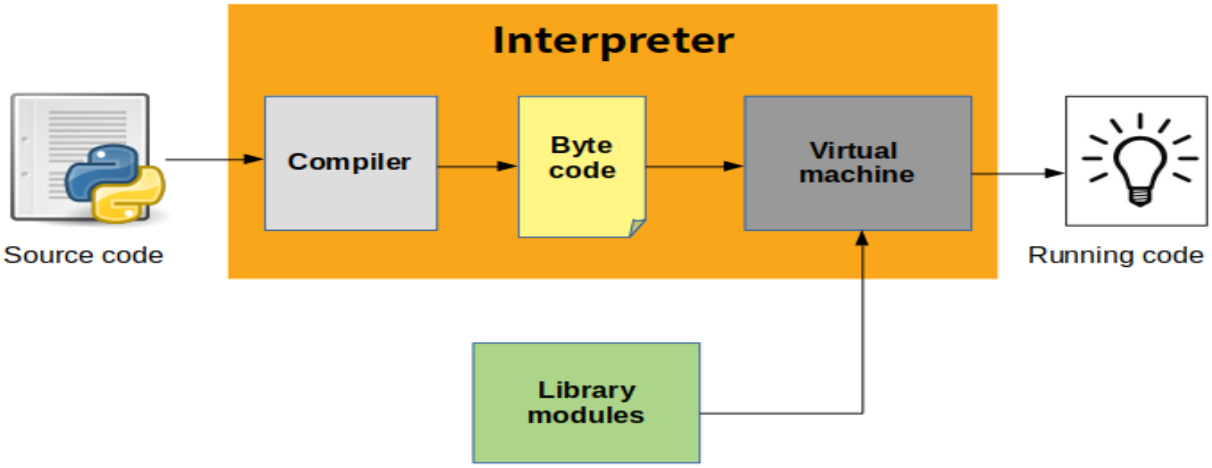
Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be

one and preferably only one obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the Python reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.



The Python Platform:

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.

There are four functions for getting information about the current Python interpreter. python_version() and python_version_tuple() return different forms of the interpreter version with major, minor, and patch level components. python_compiler() reports on the compiler used to build the interpreter. And python_build() gives a version string for the build of the interpreter.

Platform() returns string containing a general purpose platform identifier. The function accepts two optional Boolean arguments. If aliased is true, the names in the return value are converted from a formal name to their more common form. When terse is true, returns a minimal value with some parts dropped.

What does python technology do?

Python is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there's a framework for pretty much anything – from web apps to machine learning.

Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados.

So what are the tangible benefits the language brings to those who decided to use it as a core technology? Below you will find just some of those reasons.

PRODUCTIVITY AND SPEED

It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

PYTHON IS POPULAR FOR WEB APPS

Web development shows no signs of slowing down, so technologies for rapid and productive web development still prevail within the market. Along with JavaScript and Ruby, Python, with its most popular web framework Django, has great support for building web apps and is rather popular within the web development community.

OPEN-SOURCE AND FRIENDLY COMMUNITY

As stated on the official website, it is developed under an OSI-approved open source license, making it freely usable and distributable. Additionally, the development is driven by the community, actively participating and organizing conference, meet-ups, hackathons, etc. fostering friendliness and knowledge-sharing.

PYTHON IS QUICK TO LEARN

It is said that the language is relatively simple so you can get pretty quick results without actually wasting too much time on constant improvements and digging into the complex engineering insights of the technology. Even though Python programmers are really in high demand these days, its friendliness and attractiveness only help to increase number of those eager to master this programming language.

BROAD APPLICATION

It is used for the broadest spectrum of activities and applications for nearly all possible industries. It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence:

- Machine learning as it has a plethora of libraries implementing machine learning algorithms.
- Web development as it provides back end for a website or an app.
- Cloud computing as Python is also known to be among one of the most popular cloud-enabled languages even used by Google in numerous enterprise-level software apps.
- Scripting.
- Desktop GUI applications.

Python compiler

The Python compiler package is a tool for analyzing Python source code and generating Python bytecode. The compiler contains libraries to generate an abstract syntax tree from Python source code and to generate Python bytecode from the tree.

The compiler package is a Python source to bytecode translator written in Python. It uses the built-in parser and standard parser module to generate a concrete syntax tree. This tree is used to generate an abstract syntax tree (AST) and then Python bytecode.

The full functionality of the package duplicates the built-in compiler provided with the Python interpreter. It is intended to match its behavior almost exactly. Why implement another compiler that does the same thing? The package is useful for a variety of purposes. It can be modified more easily than the built-in compiler. The AST it generates is useful for analyzing Python source code.

The basic interface

The top-level of the package defines four functions. If you import compiler, you will get these functions and a collection of modules contained in the package.

compiler.parse(buf)

Returns an abstract syntax tree for the Python source code in buf. The function raises Syntax Error if there is an error in the source code. The return value is a compiler.ast. Module instance that contains the tree.

compiler.parseFile(path)

Return an abstract syntax tree for the Python source code in the file specified by path. It is equivalent to parse(open(path).read()).

LIMITATIONS

There are some problems with the error checking of the compiler package. The interpreter detects syntax errors in two distinct phases. One set of errors is detected by the interpreter's parser, the other set by the compiler. The compiler package relies on the interpreter's parser, so it get the first phases of error checking for free. It implements the second phase itself, and that implementation is incomplete. For example, the compiler package does not raise an error if a name appears more than once in an argument list: def f(x, x): ...

A future version of the compiler should fix these problems.

PYTHON ABSTRACT SYNTAX

The compiler.ast module defines an abstract syntax for Python. In the abstract syntax tree, each node represents a syntactic construct. The root of the tree is Module object.

The abstract syntax offers a higher level interface to parsed Python source code. The parser module and the compiler written in C for the Python interpreter use a concrete syntax tree. The concrete syntax is tied closely to the grammar description used for the Python parser. Instead of a single node for a construct, there are often several levels of nested nodes that are introduced by Python's precedence rules.

The abstract syntax tree is created by the compiler.transformer module. The transformer relies on the built-in Python parser to generate a concrete syntax tree. It generates an abstract syntax tree from the concrete tree.

The transformer module was created by Greg Stein and Bill Tutt for an experimental Python-to-C compiler. The current version contains a number of modifications and improvements, but the basic form of the abstract syntax and of the transformer are due to Stein and Tutt.

AST NODES

The compiler.ast module is generated from a text file that describes each node type and its elements. Each node type is represented as a class that inherits from the abstract base class compiler.ast.Node and defines a set of named attributes for child nodes.

class compiler.ast.Node

The Node instances are created automatically by the parser generator. The recommended interface for specific Node instances is to use the public attributes to access child nodes. A public attribute may be bound to a single node or to a sequence of nodes, depending on the Node type. For example, the bases attribute of the Class node, is bound to a list of base class nodes, and the doc attribute is bound to a single node.

Each Node instance has a lineno attribute which may be None. XXX Not sure what the rules are for which nodes will have a useful lineno.

All Node objects offer the following methods:

getChildren()

Returns a flattened list of the child nodes and objects in the order they occur. Specifically, the order of the nodes is the order in which they appear in the Python grammar. Not all of the children are Node instances. The names of functions and classes, for example, are plain strings.

getChildNodes()

Returns a flattened list of the child nodes in the order they occur. This method is like getChildren(), except that it only returns those children that are Node instances.

The While node has three attributes: test, body, and else_. (If the natural name for an attribute is also a Python reserved word, it can't be used as an attribute name. An underscore is appended to the word to make it a legal identifier, hence else_ instead of else.)

The if statement is more complicated because it can include several tests.

The If node only defines two attributes: tests and else_. The tests attribute is a sequence of test expression, consequent body pairs. There is one pair for each if/elif clause. The first element of the pair is the test expression. The second elements is a Stmt node that contains the code to execute if the test is true.

The getChildren() method of If returns a flat list of child nodes. If there are three if/elif clauses and no else clause, then getChildren() will return a list of six elements: the first test expression, the first Stmt, the second text expression, etc.

The following table lists each of the Node subclasses defined in compiler.ast and each of the public attributes available on their instances. The values of most of the attributes are themselves Node instances or sequences of instances. When the value is something other than an instance, the type is noted in the comment. The attributes are listed in the order in which they are returned by getChildren() and getChildNodes().

DEVELOPMENT ENVIRONMENTS:

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

IMPLEMENTATIONS

Reference implementation

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

Other implementations

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.

Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version.

MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes Lego Mindstorms EV3.

RustPython is a Python 3 interpreter written in Rust.

Unsupported implementations

Other just-in-time Python compilers have been developed, but are now unsupported:

Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations suffer from the global interpreter lock.

Psyco is a just-in-time specialising compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

Cross-compilers to other languages

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

- Jython enables the use of the Java class library from a Python program.
- IronPython follows a similar approach in order to run Python programs on the .NET Common Language Runtime.
- The RPython language can be compiled to C, and is used to build the PyPy interpreter of Python.
- Pyjs compiles Python to JavaScript.
- Cython compiles Python to C and C++.
- Numba uses LLVM to compile Python to machine code.
- Pythran compiles Python to C++.
- Somewhat dated Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.
- Google's Grumpy compiles Python to Go.
- MyHDL compiles Python to VHDL.

- Nuitka compiles Python into C++.

PERFORMANCE

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.

API DOCUMENTATION GENERATORS

Python API documentation generators include:

- Sphinx
- Epydoc
- HeaderDoc
- Pydoc

USES

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like Free CAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Light wave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

Python is commonly used in artificial intelligence projects with the help of libraries like Tensor Flow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

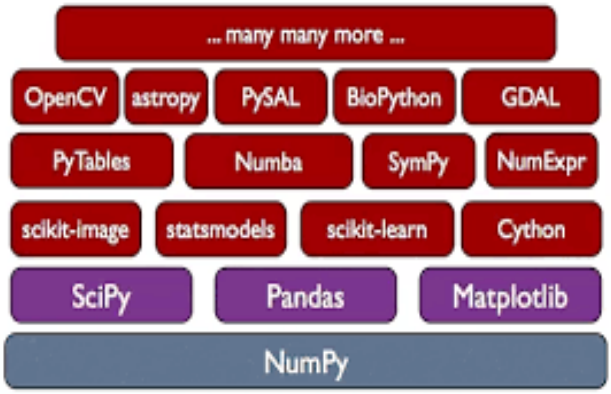
Python is used extensively in the information security industry, including in exploit development.

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.

LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.



Library features

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and sub setting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

CHAPTER 6

SYSTEM DESIGN

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

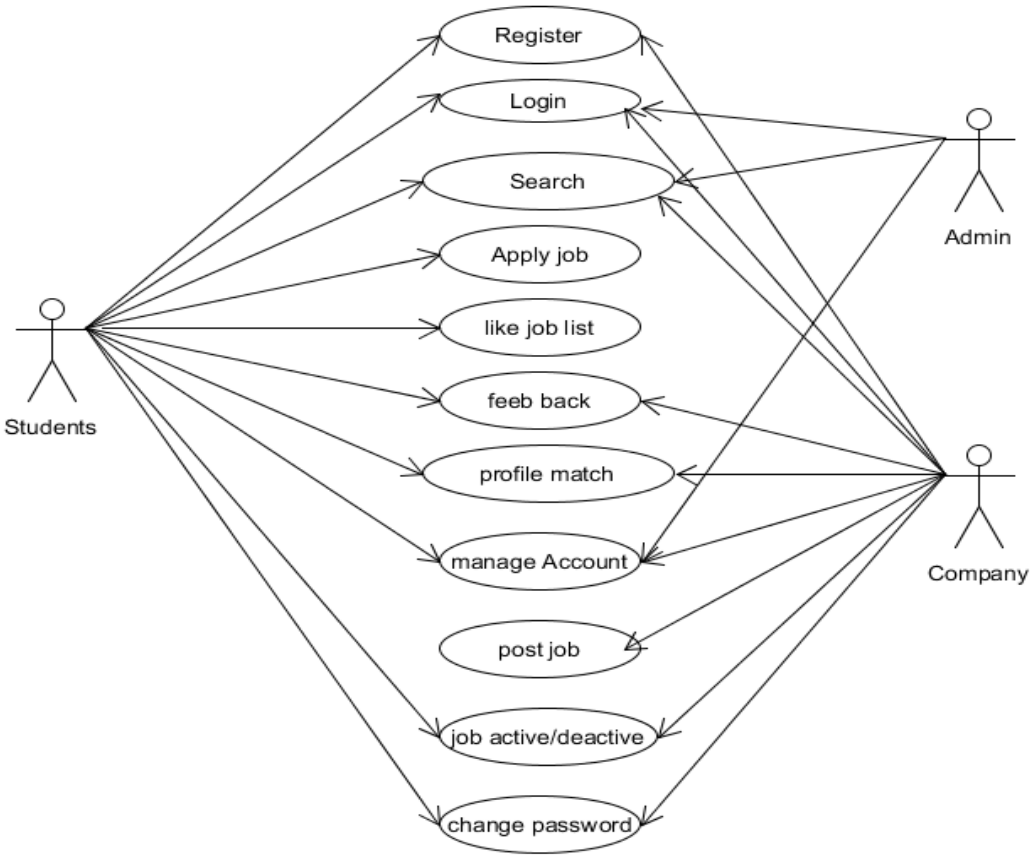
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

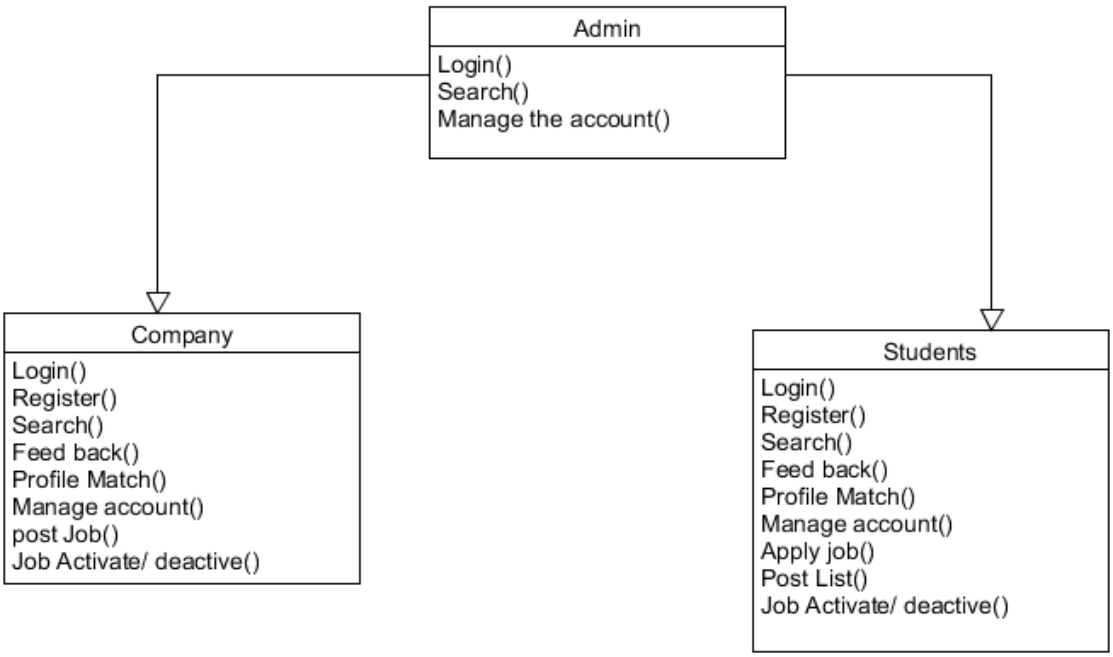
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



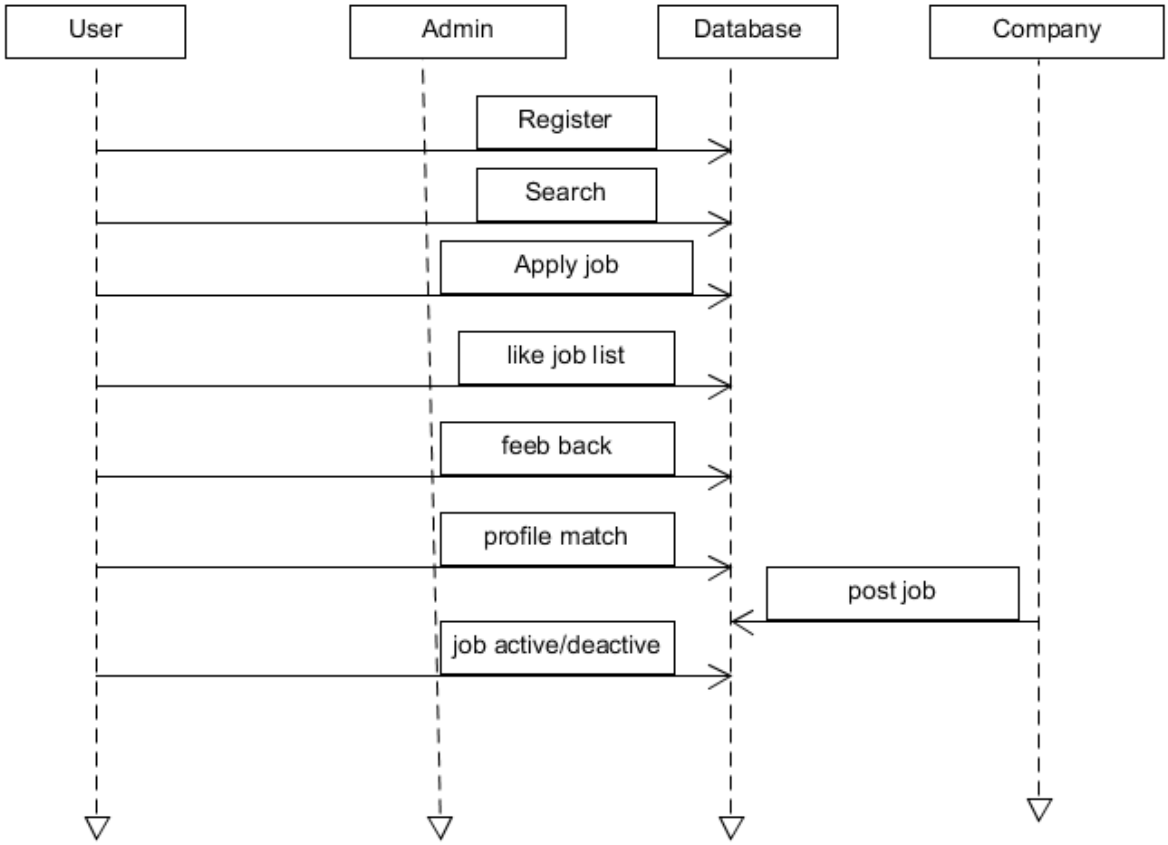
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



DEPLOYMENT:

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these

two diagrams are special diagrams used to focus on software and hardware components.

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

CHAPTER 7
SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as

shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at

least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 8

8.1 CONCLUSION

The Stress Detection System is designed to assess employee stress by reviewing photographs submitted by verified users, making the framework is reliable. After the successful registration and login, user uploads the image and also uses the live cam. After uploading the image, we will get the output of the stress level on the top of bounded box as angry, sad, happy, disgusting, and neutral.

8.2 FUTURE WORK

To identify stress, the proposed method combines image processing and deep learning. To extract features, images were gathered and analyzed. Along with the Live Cam, the video facility can also be benefitting to the future work with various algorithms. The algorithm processing outputs were used to train the model and test it with the test dataset. Despite the fact that the acquired results are preliminary due to the small number of persons involved or technical information, the key added value of this paper is acquired by permitting end - user to correctly recognize ongoing stress in order to decrease future health risk factor. A broader population study will be part of our future effort.

REFERENCE

[1] Widanti, N., Sumanto, B., Rosa, P., Miftahudin, M.F.. Stress level detection using heart rate, blood pressure, and gsr and stress therapy by utilizing infrared. In: Industrial Instrumentation and Control (ICIC), 2017 International Conference on. IEEE; 2017.

[2] Ravinder Ahujaa, Alisha Bangab " Mental Stress Detection in University Students using Machine Learning Algorithms", International Conference on Pervasive Computing Advances and Applications,2019.

[3] Sharma, L.D. , Bohat, V.K. , Habib, M. "Stress Detection Using EEG Signals", Journal of Advanced Research in Dynamical and Control Systems,2017.

[4] G. Giannakakis, D. Manousos, F. Chiarugi, "Stress and anxiety detection using facial expressions from videos," Biomedical Signal processing and Control", vol. 31, pp. 89- 101, January 2017.

[5] Nisha Raichur, Nidhi Lonakadi, Priyanka Mural, "Detection of Stress Using Image Processing and Machine Learning Techniques", vol.9, no. 3S, July 2017.

[6] U. S. Reddy, A. V. Thota and A. Dharun, "Machine Learning Techniques for Stress Prediction in Working Employees," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 2018.

[7] Reshmi Gopalakrishna Pillai, Mike Thelwall, Constantin Orasan, "Detection of Stress and Relaxation Magnitudes for Tweets", International World Wide Web Conference Committee ACM, 2018.

[8] Huijie Lin, Jia Jia, JiezhonQiu, " Detecting stress based on social interactions in social networks", IEEE Transactions on Knowledge and Data Engineering, 2017.

[9] Enrique Garcia-Ceja, Venet Osmani and Oscar Mayora, "Automatic Stress Detection in working environments from smartphones' accelerometer data: A First Step" , arXiv:1510.04221v1 [cs.HC] 14 Oct 2015.

[10] Saskia Koldijk , Mark A. Neerincx, and Wessel Kraaij, "Detecting Work Stress in offices by combining unobtrusive sensors", IEEE Transactions on affective computing 2018.

[11]Virginia Sandulescu, Sally Andrews, David Ellis, "Stress Detection using wearable physiological sensors", Springer International Publishing Switzerland 2015.

[12] Ankita Patil, Rucha Mangalekar, Nikita Kupawdekar, Viraj Chavan, Sanket Patil, and Ajinkya Yadav, "Stress Detection in IT Professionals By Image Processing And Machine Learning," International Journal of Research in Engineering, Science and Management, Vol. 3, No.1, January-2020.

[13] B. Padmaja, V. V. Rama Prasad and K. V. N. Sunitha, "A Machine Learning Approach for Stress Detection using a wireless Physical Activity Tracker," International Journal of Machine Learning and Computing, February, Vol. 8, No. 1, February 2018.

[14] Zenonos, A., Khan, A., Kalogridis, G., Vatsikas, S., Lewis, T., Sooriyabandara, "Healthy Office: Mood recognition at work using smartphones and wearable sensors " , IEEE International Conference on Pervasive Computing and Communication Workshops 2016.

[15] Tanev, G., Saadi, D.B., Hoppe, K., Sorensen, H.B "Classification of acute stress using linear and non-linear heart rate variability analysis derived from

sternal ECG" 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.

[16] Deng, Y., Wu, Z., Chu, C.H., Zhang, Q., Hsu, D.F.. Sensor feature selection and combination for stress identification using combinatorial fusion. International Journal of Advanced Robotic Systems 2013.

[17] Ghaderi, A., Frounchi, J., Farnam, A.. Machine learningbased signal processing using physiological signals for stress detection. In: 2015 22nd Iranian Conference on Biomedical Engineering (ICBME). 2015.

[18] Mrs. Megha V Gupta Dr. Shubhangi Vaikole " Recognition of Human Mental Stress Using Machine Learning Paradigms",2020.

[19] Liu, D., Ulrich, M.. Listen to your heart: Stress prediction using consumer heart rate sensors 2015. [20] Nakashima, Y., Kim, J., Flutura, S., Seiderer, A., Andre, E.. Stress recognition in daily work. In: ´ International Symposium on Pervasive Computing Paradigms for Mental Health. Springer; 2015.

[21] Xu, Q., Nwe, T.L., Guan, C.. Cluster-based analysis for personalized stress evaluation using physiological signals. IEEE journal of biomedical and health informatics 2015.

[22] Gjoreski, M., Gjoreski, H., Lustrek, M., Gams, M.. Continuous stress detection using a wrist device: in laboratory and real life. In: Proceedings of the 2016 ACM International Joint Conference on www.jespublication.com Pervasive and Ubiquitous Computing: Adjunct. ACM; 2016.

[23] Palanisamy, K., Murugappan, M., Yaacob, S.. Multiple physiological signal-based human stress identification using non-linear classifiers.Elektronika ir elektrotechnika 2015.

[24] Villarejo, M.V., Zapirain, B.G., Zorrilla, A.M.. A stress sensor based on galvanic skin response (gsr) controlled by zigbee. Sensors,2012.

[25] Selvaraj, N.. Psychological acute stress measurement using a wireless adhesive biosensor. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.

[26] Sriramprakash.S ∗, Prasanna Vadana. D, O. V. Ramana Murthy "Stress Detection in Working People" 7th International Conference On Advances In Computing & Communications, ICACC 2017, August 22-24, 2017.

[27]Koldijk, S., Neerincx, M.A., Kraaij, W.. Detecting work stress in offices by combining unobtrusive sensors. IEEE Transactions on Affective Computing 2016.

SOURCE CODE

```
from flask import Flask, request, jsonify,render_template

import pickle

import pandas as pd

import joblib

app = Flask(__name__)


@app.route('/')

def home():

    return render_template('admin.html')


@app.route('/index')

def index():
```

```
    return render_template('index.html')


@app.route('/adminval',methods=['POST','GET'])
def adminval():
    if request.method == 'POST':
        uname = request.form.get('username')
        upass = request.form.get('password')
        if uname == 'admin' and upass == '1234':
            return render_template('index.html')
        else:
            return render_template('admin.html', msg = 'Invalid Data')


@app.route('/predict', methods=['POST','GET'])
def predict():
    input_features = []
    age = request.form.get('age')
    gender = request.form.get('gender')
    dsg = float(request.form.get('dsg'))
    ra = float(request.form.get('ra'))
    mfs = float(request.form.get('mfs'))
    wfh = request.form.get('wfh')
    com = request.form.get('com')
```

```
    input_features.append(dsg)
    input_features.append(ra)
    input_features.append(mfs)
    loaded_model = joblib.load('model.joblib')

    prediction = loaded_model.predict([input_features])
    msg1 = prediction[0] * 100
    msg = "The stress level is "  + str(msg1) + " %"
    return render_template('index.html', msg=msg)



if __name__ == '__main__':
    app.run(debug=True)
```

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Admin</title>
        <meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
<style>
a{text-decoration:none;color: #5e5776;}
body{background-image: url('static/img/f-bg.jpg');background-repeat:no-repeat;background-size: cover;
background-position: center;}

button{border:none;cursor:pointer;height:35px;width:100px;border-radius:2em;background-color:#5e5776;
margin-left:100px;color: #fff;}
.mainForm{padding:130px 50px;}
.logform{text-align:center;height:370px;width:700px;padding-top:60px;margin:auto;background:#d9d0ff;border-radius:25px;}
input{outline:none;width:45%;height:35px;padding:5px;}
.inp,h3{margin-bottom:30px;}
.resPage{margin-top:10px;margin-left:100px;}
label{width:120px;}
</style>
</head>
```

```
<body>
    <div class="mainForm">
        <form class="logform" action="/adminval" method="POST">
            <h3>ADMIN LOGIN</h3>
            <div class="inp">
                <label>Username : </label>
                <input type="text" name="username">
            </div>
            <div class="inp">
                <label>Password : </label>
                <input type="password" name="password">
            </div>
            <button type="submit">Login</button>
            <div class="resPage">
                <p>{{msg}}</p>
            </div>
        </form>
    </div>
</body>
</html>
```

```
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
  <title>Stress Detection - Homepage</title>
</head>
<body class="bg-nav">
<nav class="navbar">
  <a href="#" class="navbar-brand text-light" style="margin-left: 20px">Stress Detection</a>
</nav>
<div class="container">
<!--  <div class="row">-->
```

```
<!--    <div class="col-md-8">-->
<!--    <h1 class="text-light display-4 mt-100" style="font-size:80px">ABCDEF TEXT</h1>-->
<!--    </div>-->
   <div class="col-md-12">
     <h1 align="center"> STRESS LEVEL PREDICTION</h1>
     <div class="card mt-100">

       <div class="card-body">

         <form class="form" action="/predict" method="POST">

           <label for="age">Age</label><br>
           <input type="number" class="form-control" name="age" id="age" min="1" max="100" placeholder="(1, 100)" step="any" required><br>

           <label for="sex">Gender</label><br>
           <input type="number" class="form-control" name="gender" id="sex" min="0" max="1" placeholder="Male - 1, Female - 0" required><br>

           <label for="tsh" >Designation Level</label><br>
           <input type="number" class="form-control" name="dsg" id="tsh" placeholder="Designation Level" step="any" required ><br>
```

<label for="t3">Resource Allocation</label>

<input type="number" class="form-control" name="ra" id="t3" placeholder="Resource Allocation" step="any" required>

<label for="t4u">Mental Fatigue Score</label>

<input type="number" class="form-control" name="mfs" id="t4u" placeholder="Mental Fatigue Score" step="any" required>

<label for="sick">WFH Setup Available</label>

<input type="number" class="form-control" name="wfh" id="sick" min="0" max="1" placeholder="Yes - 1, No - 0" required>

<label for="pregnant">Company Type</label>

<input type="number" class="form-control" name="com" id="pregnant" min="0" max="1" placeholder="Product - 1, Service - 0" required>

<!-- <input type="submit" class="btn btn-primary btn-block btn-lg" value="Predict">
-->

<input type="submit" name="sb" value="Predict" class="sub btn btn-primary"></p>

```
                <h2 align="center">{{ msg }}</h2>
            </form>

<!--                <br><center><h2 style="color: blue"> {{prediction_text}}
</h2></center><br>-->
        </div>
      </div>
    </div>
</div>


    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.j
s"                                                    integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbT
ov4+1p" crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.
js"                                                    integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1
zdB" crossorigin="anonymous"></script>
```

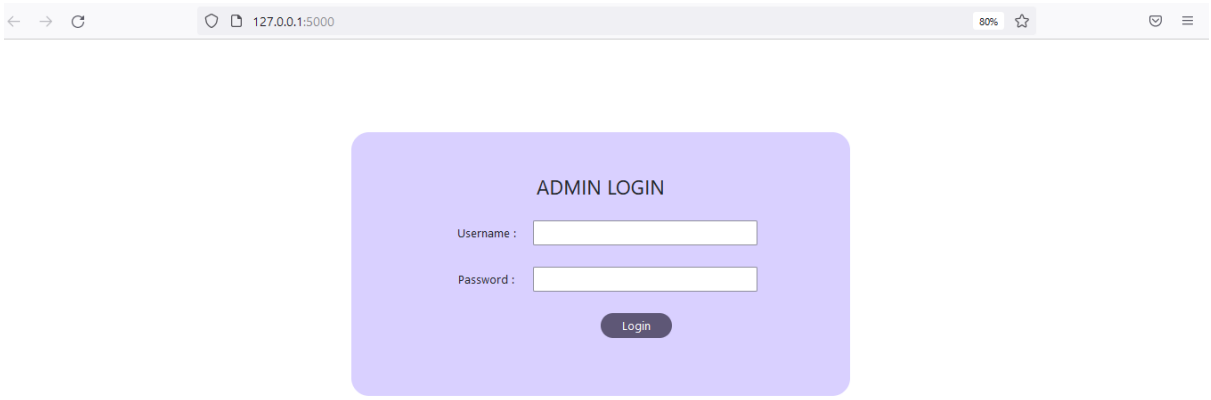
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

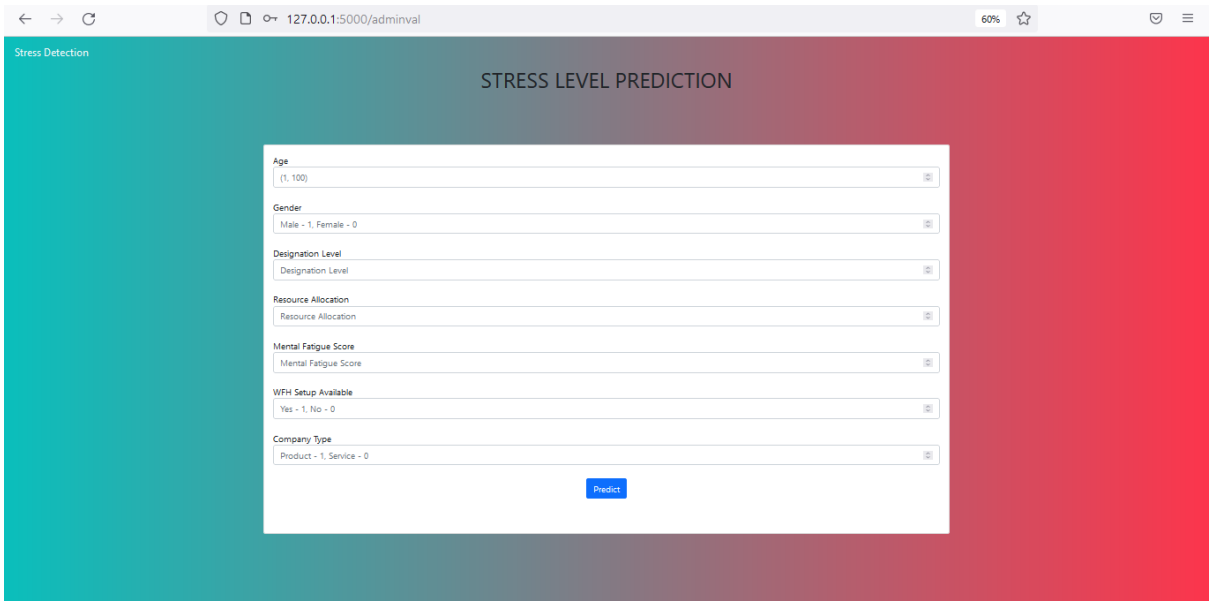
-->

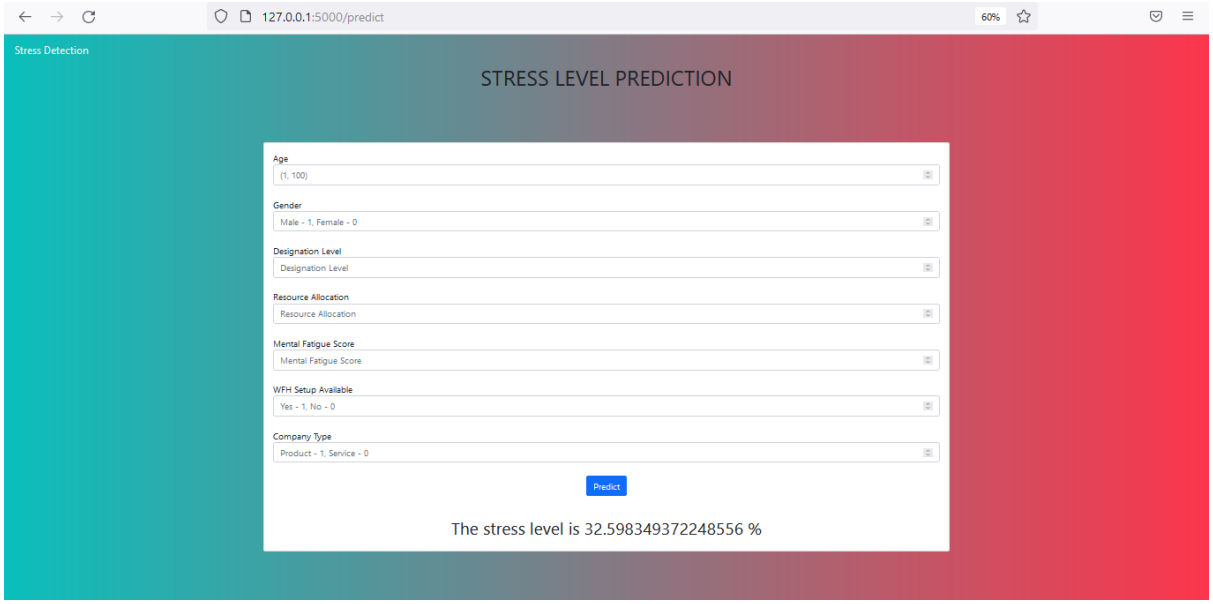
</body>

</html>

SCREENSHOT:


127.0.0.1:5000
80%
ADMIN LOGIN
Username :
Password :
Login


127.0.0.1:5000/adminval
60%
Stress Detection
STRESS LEVEL PREDICTION
Age
(1, 100)
Gender
Male - 1, Female - 0
Designation Level
Designation Level
Resource Allocation
Resource Allocation
Mental Fatigue Score
Mental Fatigue Score
WFH Setup Available
Yes - 1, No - 0
Company Type
Product - 1, Service - 0
Predict


127.0.0.1:5000/predict
60%
Stress Detection
STRESS LEVEL PREDICTION
Age
(1, 100)
Gender
Male - 1, Female - 0
Designation Level
Designation Level
Resource Allocation
Resource Allocation
Mental Fatigue Score
Mental Fatigue Score
WFH Setup Available
Yes - 1, No - 0
Company Type
Product - 1, Service - 0
Predict
The stress level is 32.598349372248556 %