Employee Performance Evaluation Model using Machine Learning Classifiers and a Virtual Voice Assistant for Routine Tasks

Revati Pawar*[1] , Devansh Mishra[2], Kritika Javali[3] , and Prof. Shruti Danve[4]

[1]*Electronics and Communication Engineering, MIT World Peace University, Pune, India*
[2]*Electronics and Communication Engineering, MIT World Peace University, Pune, India*
[3]*Electronics and Communication Engineering, MIT World Peace University, Pune, India*

[4]*Electronics and Communication Engineering, MIT World Peace University, Pune, India*

Abstract: *This paper focuses on a conversational voice assistant for organizations that helps employees accelerate common time-consuming tasks, enhancing productivity and efficiency in the process. Speech Recognition and TTS techniques allow them to set leaves, manage conference rooms, schedule meetings, and update work hours with simple voice commands. This study also takes into consideration the importance of performance analysis of employees, and this is achieved with the help of machine learning classification models namely L1 and L2 Regularization, Random Forest, and Gradient Boost. The experimental results showcase that the highest accuracy, F1 Score, Train and Test AUC-ROC is obtained via the Random Forest Classification Algorithm with values being 97.08%, 98.27%, 99.82%, and 98.41% respectively.*

Keywords: **Speech recognition, machine learning, voice assistant, employee performance, prediction**

1. Introduction

Smart systems are highly in demand due to the efficient and interactive experience they bring to any organization. Artificial intelligence has enabled individuals and organizations to implement technologies like natural language processing, machine learning, and speech recognition to enhance their experience as users by making their routine tasks interactive [1]. Alexa, Google Assistant, and Cortana are an example of how voice-enabled bots can transform the way humans interact with machines. Organizations have identified this and are moving towards practices that would increase efficiency as well as analyse an employee's ability to remain productive for continuous periods. Routine tasks kill productivity and creativity in the office environment. Every employee in an organization including the manager must perform small yet multiple time-consuming, monotonous tasks like scheduling a meeting with another employee, applying for leaves, setting up conferences, and logging daily/weekly time sheets that can take up unnecessary time while interrupting their workflow. Analysing employee performance manually is a lengthy process for the organization and hence, time-consuming. Such tasks cannot be automated but can surely be streamlined and centralized, reducing their overall time consumption. This is precisely what we strive to achieve through our implementation. This paper

*Corresponding Author

proposes a smart virtual assistant that can do all the above with simple voice commands and machine learning techniques that can give useful insights and analyse the performances of employees. A virtual assistant is an artificial intelligence-based technology. The programme receives verbal requests through the device's microphone, with the voice output coming from the speaker. But it is what happens in between these two activities that is the most fascinating. This paper aims towards streamlining and performing these specific tasks using voice commands to mitigate time loss as well as analyse the employee's performance.

2. Methodology

The user will be interacting via a voice-enabled assistant embedded into a Graphical User-Interface. Credential-based authorization is used to differentiate between a Manager and an Employee providing different access levels to each. To implement the functionalities and maintain a record of the actions performed, MySQL is used to create a database of the organization which has different functionalities sketched out for the employees and the manager. Our approach is unique as all the functionalities are enabled with speech recognition and the employees can converse with the voice assistant, making it interactive and enhancing the experience of the user in the process. The overall study aimed at achieving this can be split into three broad categories, namely the database and the schemas, speech recognition for the voice-controlled functionalities, and the employee performance machine learning model.

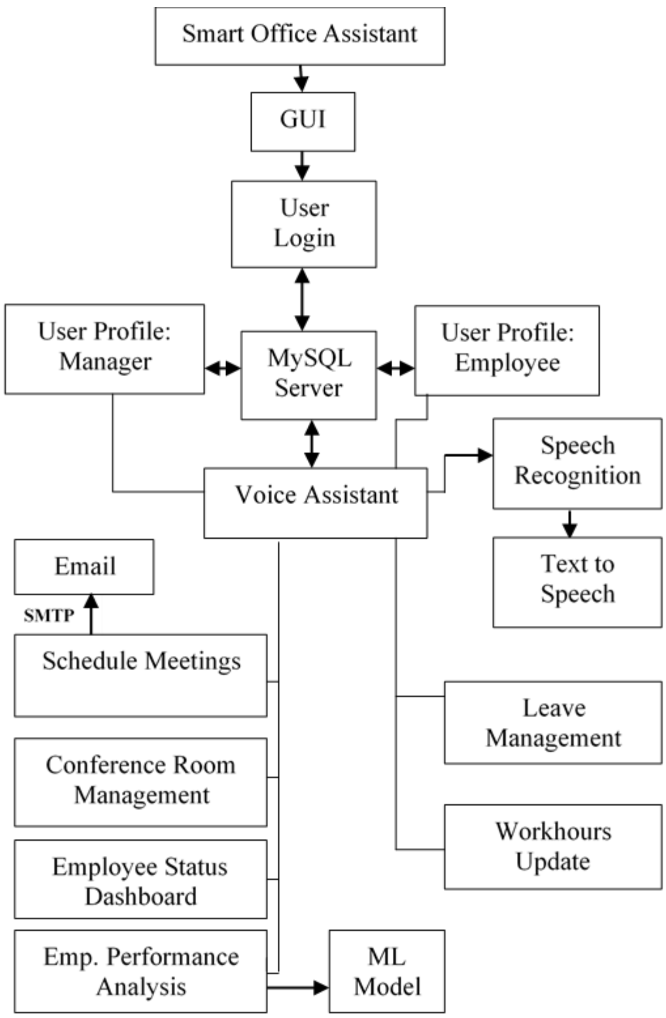


Figure 1. System Architecture Overview

2.1. MySQL Database

MySQL is the most widely used database management system software for managing relational databases today. Keeping the scope of our implementation of the virtual assistant restricted but not limited to a small organization, we have an employee database monitored by a manager. The database includes different tables for fulfilling the functionalities supported. The schema for the organization's database has been shown in Figure 2 and Figure 3. The former assists with storing and retrieving data for setting leaves, scheduling meetings, updating work hours, and booking conference rooms, while the latter supports retrieving data of each employee along with their job roles and parameters required to pass to the classification model.

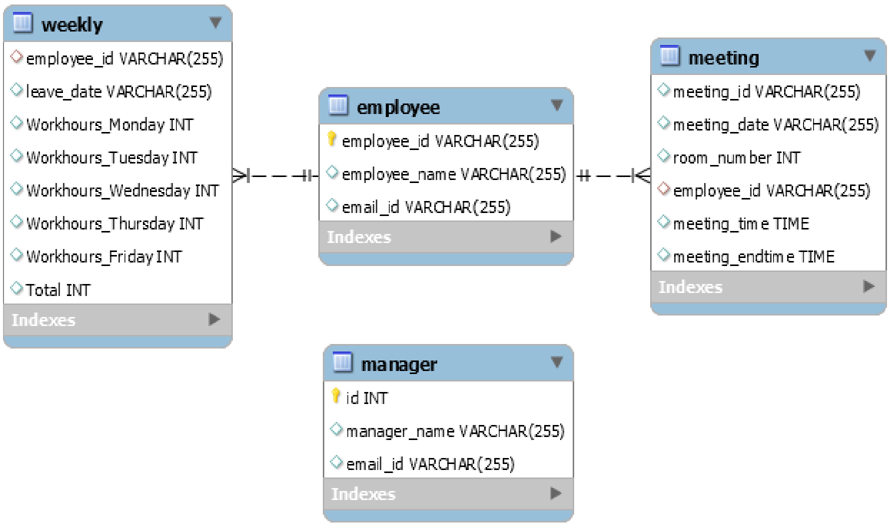


Figure 2. Schema for Database 1

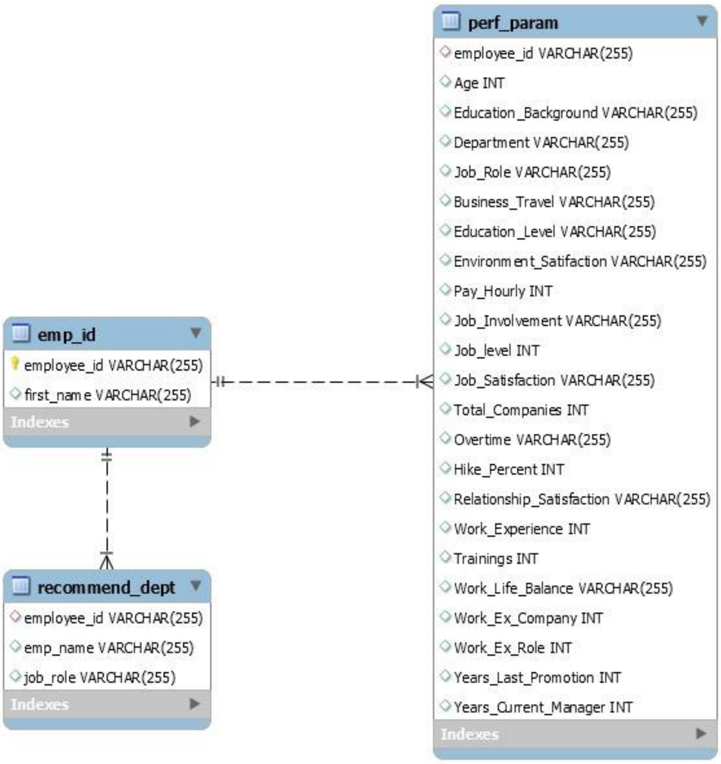


Figure 3. Schema for Database 2

2.2. Speech Recognition

The very first step in building a voice assistant is taking the speech as input. The programming language used throughout our implementation is Python. Speech Recognition involves converting physical sound into electrical signals with the help of a microphone and then further into digital data. We have used PyAudio which helps in recording audio through the microphone, the Speech Recognition library, and the Google Speech API to take the speech input and perform the speech-to-text conversion. Automatic Speech Recognition lies at the core of processing this speech data. The ASR first recognizes the speech input through the microphone, converts it into a file, removes background noise, normalizes volume, breaks down the sentences, and then implements statistical probabilities to find out the words before converting them into text [2]. After speech-to-text is complete, the next step is text-to-speech conversion. The user gives a command, and the voice assistant performs the required action based on the list of functionalities and gives confirmation to the user or returns the required data. To convert the acknowledgement or the data fetched by the voice assistant into speech, we have used the pyttsx3 library on Windows, which is supported by the Sapi5 driver. For Mac OS, the NSSpeechSynthesizer is used and espeak is used for all other platforms. Pyttsx3 allows us to choose a voice for our assistant between male and female. The login page for the user to access any functionality is shown in Figure 4.

Figure 4. Login Page

The functionalities that can be performed by the voice assistant are explained in detail below:

2.2.1 Meeting Scheduler: The manager can schedule meetings with voice commands which recognizes the date, start time, and end time of the meetings. It also takes in the ID of the employee with whom the manager wants to set up a meeting. To avoid clashes in timings, our voice assistant ensures the other employee is free and has no other meeting scheduled and responds with a confirmation or asks the manager to schedule for some other time. The other employee is notified about the meeting details via email. We have used the SMTP (Simple Mail Transfer Protocol) to send emails. It involves setting up a connection, transferring the mail data and then terminating the connection [3]. To set up a secure connection and protect the user's credentials, Secure Sockets Layer (SSL) is another protocol used to encrypt our connection to make sure no data is leaked or prone to hacking.

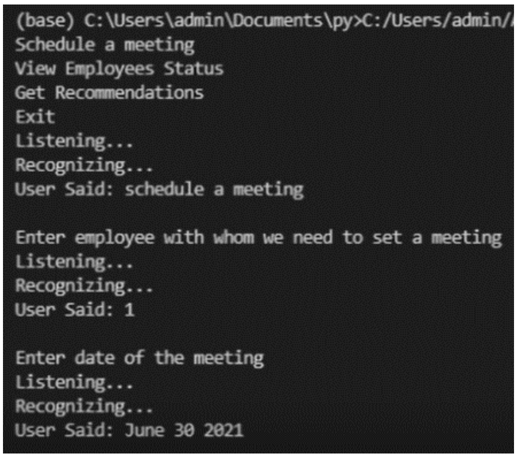


Figure 5. Meeting Scheduler (a)

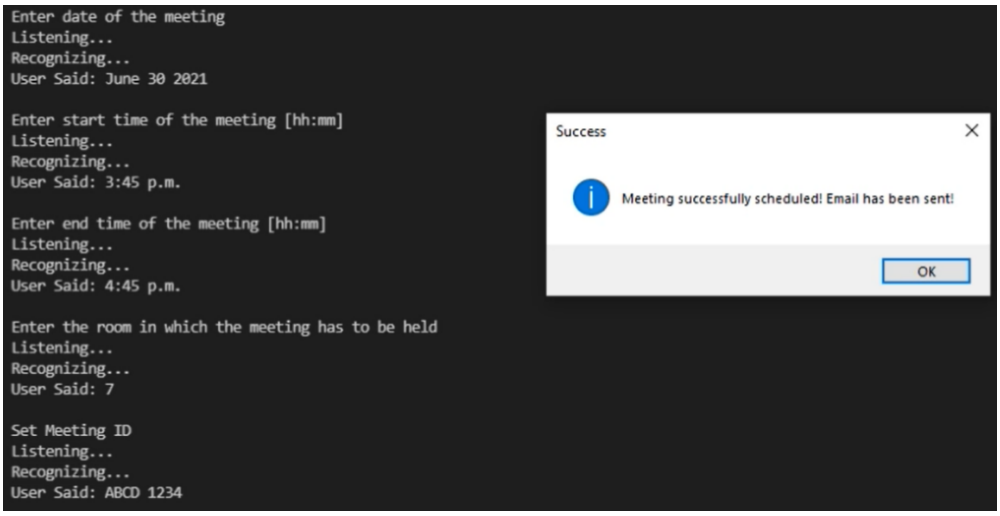


Figure 6. Meeting Scheduler (b)

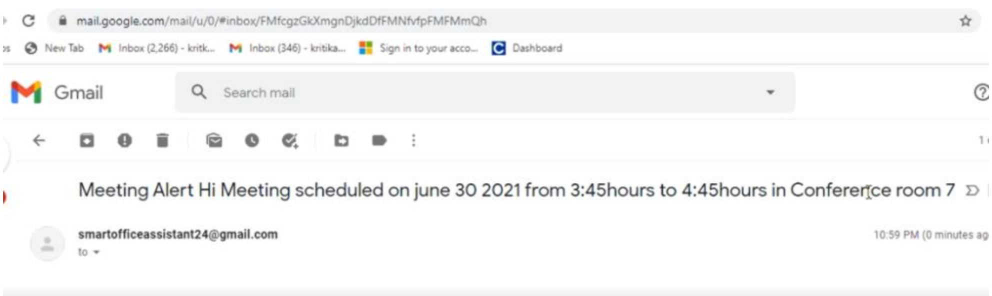


Figure 7. Email Sent via SMTP for the Meeting

2.2.2 Applying for Leaves: Employees can apply for leaves through voice commands. The meeting scheduler takes into consideration this aspect as well. If an employee is on leave and another employee tries to schedule a meeting, they are instantly notified about it and therefore the manager can reschedule to some other date.

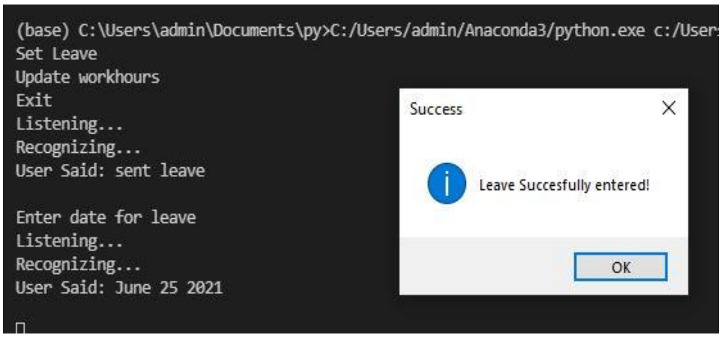


Figure 8. Leave Application

2.2.3 Updating Workhours: Filling timesheets and entering the number of hours of work is a tedious repetitive task that our voice assistant accomplishes with a single voice command.

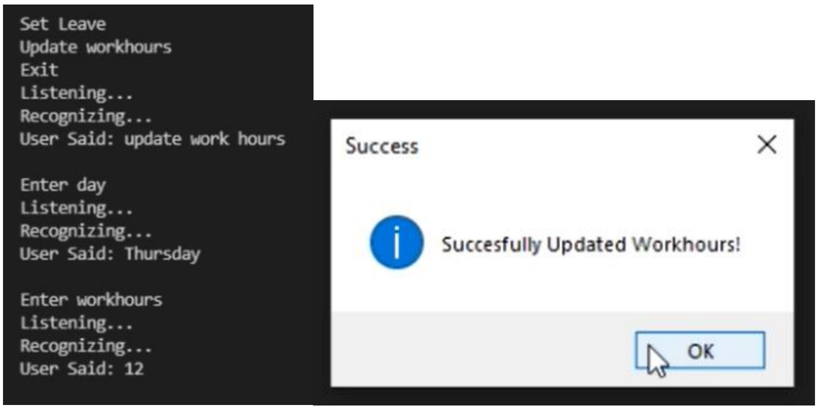


Figure 9. Workhours Update

2.2.4 Conference Room Management: In case the manager needs to book a conference room for a meeting, they can give a voice command and book the same. The assistant lets the manager know if the room is booked within the time frame or if it is free and gives a confirmation accordingly. This ensures no two meetings clash and book the same conference room.

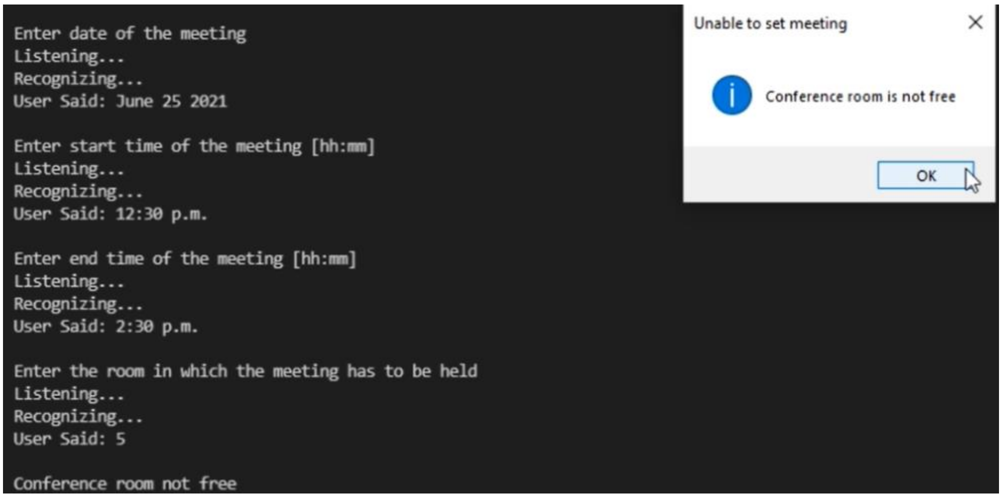
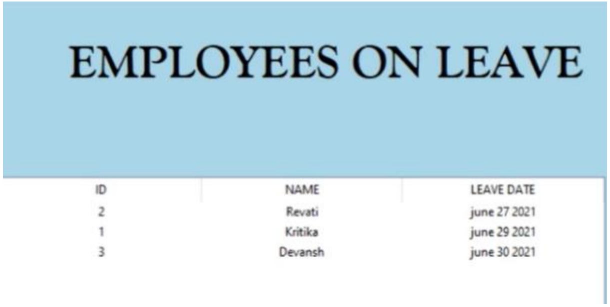


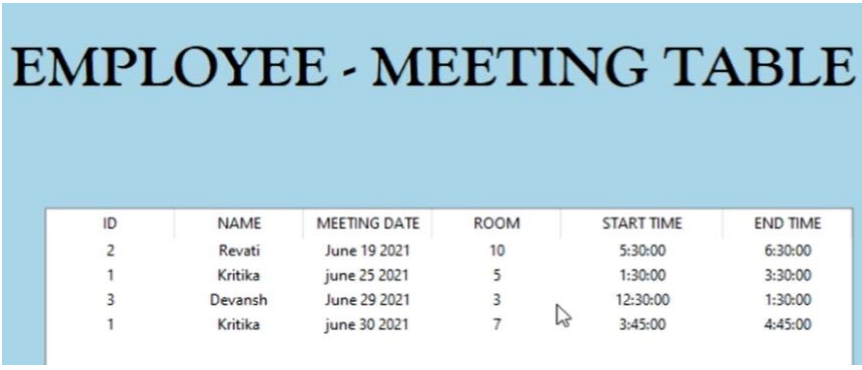
Figure 10. Conference Room Conflict

2.2.5 Dashboard: The manager gets the privileges of accessing all the above data with the help of a dashboard that the voice assistant fetches with a voice command. It details all the leaves applied for, all meetings scheduled at different conference rooms, and the work hours of all employees. The privileges granted to the manager also include the functionality of analysing the performance of the employees with a classification model to make better-informed decisions.


EMPLOYEES ON LEAVE

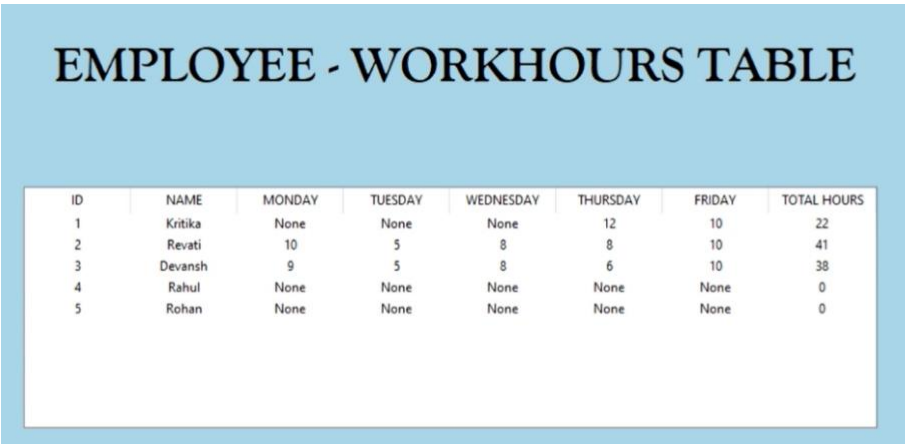
ID	NAME	LEAVE DATE
2	Revati	june 27 2021
1	Kritika	june 29 2021
3	Devansh	june 30 2021

Figure 11. Leave Dashboard


EMPLOYEE - MEETING TABLE

ID	NAME	MEETING DATE	ROOM	START TIME	END TIME
2	Revati	June 19 2021	10	5:30:00	6:30:00
1	Kritika	june 25 2021	5	1:30:00	3:30:00
3	Devansh	June 29 2021	3	12:30:00	1:30:00
1	Kritika	june 30 2021	7	3:45:00	4:45:00

Figure 12. Scheduled Meetings Dashboard


EMPLOYEE - WORKHOURS TABLE

ID	NAME	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	TOTAL HOURS
1	Kritika	None	None	None	12	10	22
2	Revati	10	5	8	8	10	41
3	Devansh	9	5	8	6	10	38
4	Rahul	None	None	None	None	None	0
5	Rohan	None	None	None	None	None	0

Figure 13. Employee Workhours Dashboard

2.3. Employee Performance Analysis Classification Model

For this study, we used an HR dataset [4] which consists of employee records for various parameters as listed in Table 1. Figure 4 showcases the algorithm for the process used in building the models. The data pre-processing stage involves cleaning and filling in missing values and removing duplicates. This is a crucial step as missing entries can lead to the algorithm not performing well or failing. The data is visualized to understand the relationship between all variables with the help of heatmaps. On analysing the correlation between the Target Variable (Performance Rating) and the other parameters, the top factors were found to be the Environment Satisfaction, Salary Hike Percent, Work-Life Balance, Years in Current Role, Years since Last Promotion, and Years with the Current Manager. The first three have a positive correlation indicating that the greater the values, the better the performance. The remaining three have a high negative correlation which indicates that they have an inverse relationship with the target variables. To gain a better understanding, we assumed the hypothesis that the latter three parameters translate into stagnated growth of an individual due to poor performance. An important factor for growth in a company is a promotion. If an employee has been in the same role for a relatively long duration, it would mean they haven't had a promotion. Hence, supporting the hypothesis. After this stage, the dataset was split into a training set (80%) and a testing set (20%), hyperparameter tuning was performed and pipelines of the four classification models were created. The task of determining a subset of optimum hyperparameters for a learning algorithm is known as hyperparameter optimization or tuning in machine learning. A hyperparameter is a value for a parameter that is used to influence the learning process [5]. The classification models were trained and the performance of each was evaluated using various performance metrics namely Accuracy, Precision, Recall, and F1-Score. Finally, the model with the best performing algorithm is chosen to predict and analyse the performance of employees.

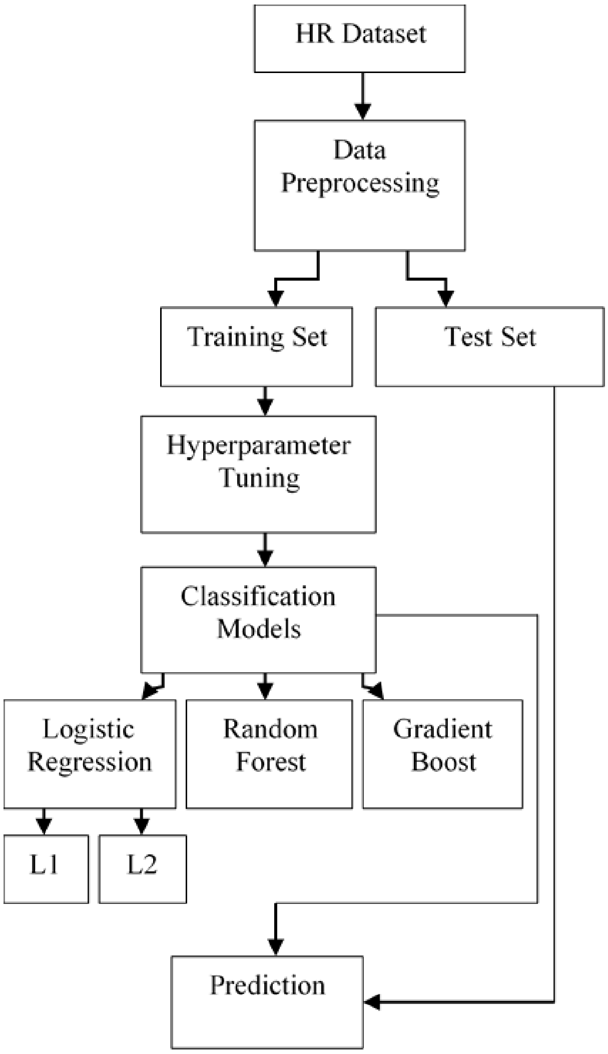


Figure 14. Machine Learning Classifier Overview

Table 1. HR Dataset

Feature	Datatype
Age	Numeric
Education Backgrond	Categorical
Employee Department	Categorical
Job Role	Categorical
Business Travel	Categorical
Education Level	Categorical
Environment Satisfaction	Categorical
Pay Scale	Numeric
Job Involvement	Categorical
Job Level	Categorical
Job Satisfaction	Categorical
Total Companies Worked	Numeric
Overtime	Categorical
Last Salary Hike Percent	Numeric
Relationship Satisfaction	Categorical
Total Work Experience	Numeric
Trainings	Numeric
Work Life Balance	Categorical
Experience at Current Company	Numeric
Experience in Current Role	Numeric
Years Since Last Promotion	Numeric
Years with Current Manager	Numeric
Performance Rating	Categorical

In this study, all algorithms are implemented along with the grid search method which is helpful in determining the optimal classifier parameters that allow a model to accurately predict certain unlabelled data. Some hyperparameters that cannot be learned directly from the training process are tuned using the Grid Search approach. There are numerous hyperparameters in the classification model and determining the ideal combination of these values is a difficult task. The Grid Search method is one of the most effective approaches for this purpose.

2.3.1 Random Forest: This is a popular classification technique used due to its reliability and it produces great results. It’s a bag of decision trees with multiple hyper parameters which when trained is like training multiple models and finally arriving at a more accurate prediction [6]. The hyperparameters defined are several estimators that determine the number of decision trees, the maximum depth which is the length of the path that’s longest from the tree root to a leaf [7], and maximum features. Greater depth ensures that more combinations are taken into consideration and hence more information is analysed. This contrasts with the effect greater depth has on decision trees alone where overfitting is directly proportional to the depth. However, in the case of random forests, this is not an issue.

2.3.2 Gradient Boosting Algorithm: This is an ensemble of decision trees that can be used to predict the target variable or classify it. The key difference between random forest and gradient boost algorithms is that the former merges all decision trees and arrives at the result at the end whereas, gradient boost performs this action after every decision tree [8]. Results and analysis have proven random forests to perform better with datasets with noise whereas, gradient boost can be difficult to tune for the same [8]. The hyperparameters defined in our model are the number of estimators and maximum depth which are similar to the ones defined in Random Forest Hyperparameters. The learning rate is the rate at which the model learns. As each tree is modified before moving on to the next, the learning rate is a key determinant in the magnitude of the modification [8].

2.3.3 Logistic Regression: This is a classification technique often used as a base model. There are two regularization techniques used to prevent overfitting: Lasso or L1 Regularization and Ridge or L2 Regularization. The two can be differentiated based on the penalty term. By adding the absolute value of weight parameters to the cost function, L1 regularisation adds the penalty term, whereas L2 regularisation adds the squared value of weights to the cost function.

The performance metrics used in this study have been elaborated below. A few terminologies used throughout the equations are True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP are the total number of positive predictions which are actually positive. FP are cases where prediction is positive, but the actual output is negative. TN are cases opposite to that of TP, where the prediction and actual values are both negative. FN are cases where the prediction is negative, but the actual output is positive.

2.3.4 Accuracy: Accuracy is defined as the percentage of correct predictions out of all the data points considered [9]. It can be calculated easily by dividing the number of predictions that are accurate by the number of total predictions as seen in Equation (1).

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{1}$$

2.3.5 Precision: As seen in Equation (2), it is the ratio of the total true positives to all the examples predicted as positive [9].

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

2.3.6 Recall: It is the ratio of examples that were true positives to all the examples that truly belonged to the said class [9]. This includes the false negatives as seen in Equation (3).

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

2.3.7 F1-Score: As can be seen in Equation (4), this is a combination of Precision and Recall. It is the harmonic mean of the two [9].

$$F1\ Score = 2 \times \frac{Precision \times Recall}{(Precision+Recall)} \tag{4}$$

2.3.8 AUC-ROC Curves: ROC is a plot showcasing the ability of a binary classification model to distinguish between two classes with the threshold of discrimination varying. The AUC is the plot of False Positive Rate (FPR) vs. True Positive Rate (TPR). The TPR is also known as Sensitivity or Recall, which is already seen in Equation (3). The FPR is the proportion of negative data points that are wrongly classified as positive when applied to all negative data points as seen in Equation (5). This method is popularly used for binary classification techniques [10] as it gives a measure of how well the model distinguishes between different classes. The AUC measures how well the model predicts 0s and 1s. Higher AUC is an indication of better performance.

$$FPR = \frac{FP}{TN+FP} \quad (5)$$

3. Results and Discussion

The results from the office assistant's performance are satisfactory and the voice recognition is accurate, prompt, and has a quick response rate. The user interface integrates the login page, voice assistant and functionalities, and the performance analysis classification model that enables a smooth experience and easy accessibility for the user. All functionalities are successfully carried out, with the database synced at all points of time and therefore, making our voice assistant efficient and accessible. The results obtained from the four classification algorithms are summed in Table 2. From Table 2, we can conclude that the Random Forest Classifier has high Accuracy, Precision, Recall, F1 Score, and a high percentage for both the test and training set ROC curve. The dataset used was imbalanced and hence, relying on accuracy alone can give us biased results. To get an accurate analysis comparison of the models' performance on both training and test sets, we plotted Receiver Operating Characteristic (ROC) curves to find out the Area Under Curve (AUC). The Area Under Curves represented by Figure 15, 16, 17, and 18 is plotted between FPR and TPR telling us how capable the logarithm is in distinguishing the classes. Hence, the model with the highest AUC for both the Train Set and Test Set is the Random Forest Classifier at 99.82% and 98.41% respectively. This means that this algorithm is better than other algorithms at distinguishing between good and poor performers.

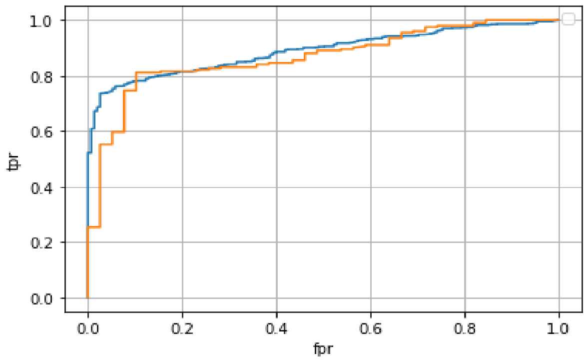


Figure 15. ROC L1 Regularization (FPR vs. TPR)

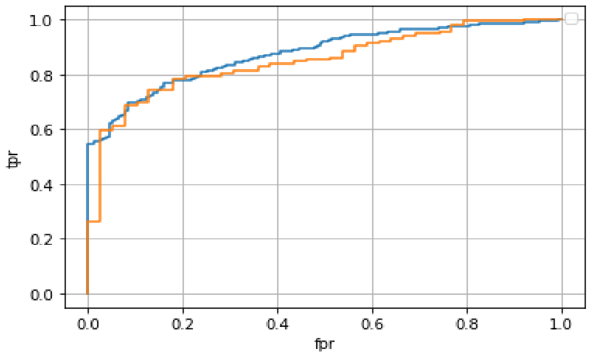


Figure 16. ROC L2 Regularization (FPR vs. TPR)

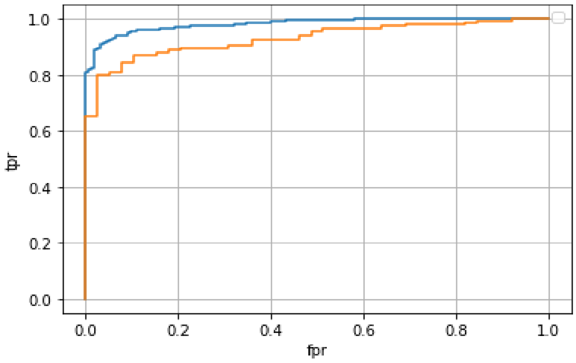


Figure 17. ROC Random Forest (FPR vs. TPR)

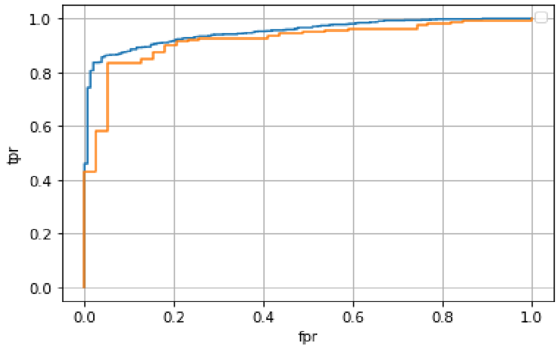


Figure 18. ROC Gradient Boost (FPR vs. TPR)

Table 2. Results

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Train AUC (%)	Test AUC (%)
Lasso (L1)	87.08	90.86	94.02	92.42	91.94	90.99
Ridge (L2)	85.83	88.12	96.01	91.90	91.07	89.96
Random Forest	97.08	97.54	99.00	98.27	99.82	98.41
Gradient Boosting	96.66	97.53	98.50	98.01	99.66	98.22

The high precision is an indicator of the results being more consistent and high recall is an indicator of more relevant results. Given f1 score is a combination of both precision and recall, it means that our model has lesser false positives and false negatives which translates into a good accuracy of the model. Therefore, we can conclude that we have achieved the best results from Random Forest Classifier closely followed by the Gradient Boosting Algorithm, the accuracy of both being 97.08% and 96.66% respectively. A few insights that can be drawn from the model are that employees with a high environment satisfaction have a strong relationship with good performance and hence, workspaces can focus on improving this factor. Salary hikes influence the employee's performance as it leads to a financial and psychological boost. The years since the last promotion have an inverse relationship with performance as promotion is the result of a good performer. Therefore, the organization head or the HR department can gain insights from data to make decisions. In case an employee's performance falls in the poor category, the organization can intervene and analyse where the employee is falling short. It can be due to lack of training, poor environment satisfaction, poor work-life balance, or even due to low job involvement. If an employee falls in the good performing category, they can be offered salary hikes or promotions to ensure the level of performance is not just maintained but improved as well.

4. Conclusion

In our study and implementation, we have taken a unique approach to making day-to-day redundant activities in workspaces more interactive. Conversational bots are transforming the way organizations work and that was the source of our inspiration. The speech recognition modules and text-to-speech conversions give a hassle-free experience. The Employee Performance model got the best results from Random Forest Classifier with an accuracy of 97.08 and an F1 score of 98.27. Previous work for solving the employee performance classification has been implemented using different methods with one of the models being the Naïve Bayes Classifier with an accuracy of 95.48 [11]. Our study using Random Forest generates an accuracy greater than the previously known by 1.6%, with a difference in the features and dataset attributes. It can be deployed in workspaces which will help make decisions that will help the organization grow and progress. In conclusion, this study demonstrates how human resources are critical to an organization's growth and what measures can be taken to enhance an employee’s experience and make workspaces more interactive by introducing voice assistants. The human resource department may determine whether the employee will adhere to the business requirements by analyzing performances using machine learning models and focusing on improvements and celebrating successes. Virtual assistants have the potential to become trusted collaborators in corporate processes by utilizing AI tools' scientific merit.

5. References

[1] *V. Këpuska and G. Bohouta. Next generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 2018. DOI: 10.1109/CCWC.2018.8301638.*

[2] *S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh. Artificial Intelligence-based Voice Assistant. 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020. DOI: 10.1109/WorldS450073.2020.9210344.*

[3] *S. Kumar, Y. R. and R. Aishwarya, Voice Email Based On SMTP For Physically Handicapped, 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021. DOI: 10.1109/ICICCS51141.2021.9432206.*

[4] *Subhash, P. (2017, March). IBM HR Analytics Employee Attrition and Performance, Version 1. Available at https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset/metadata.*

[5] *Li Yang, Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, Volume 415, 2020. DOI: 10.1016/j.neucom.2020.07.061.*

[6] *Siji George C G and B.Sumathi. Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction. vol. 11, No. 9, International Journal of Advanced Computer Science and Applications (IJACSA), 2020, DOI: 10.14569/IJACSA.2020.0110920.*

[7] *El-Rayes, N., Fang, M., Smith, M. and Taylor, S.M. Predicting employee attrition using tree-based models. International Journal of Organizational Analysis, Vol. 28 No. 6, 2020 DOI: 10.1108/IJOA-10-2019-1903.*

[8] *Jerome H. Friedman, Stochastic gradient boosting, Computational Statistics & Data Analysis, Volume 38, Issue 4,2002. DOI: 10.1016/S0167-9473(01)00065-2.*

[9] *Tom M Mitchell, Machine Learning, McGraw Hill Education, 2013.*

[10] *Wang, Q, Guo, A. An efficient variance estimator of AUC and its applications to binary classification. Statistics in Medicine. 2020. DOI: 10.1002/sim.8725.*

[11] *[11] Riyanto Jayadi, Hafizh M. Firmantyo, Muhammad T. J. Dzaka, Muhammad F. Suaidy, Alfitra M. Putra. Employee Performance Prediction using Naïve Bayes, Volume 8, No.6. International Journal of Advanced Trends in Computer Science and Engineering, 2019. DOI: 10.30534/ijatcse/2019/59862019.*