Employee Attrition Prediction by Machine Learning

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled "Employee Attrition Prediction by Machine Learning" is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the "BTECH project & project report" The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Souvik Halder

Roll No. - 10330522058

Reg. No- 221030110636 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled “Employee Attrition Prediction by Machine Learning” is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the “BTECH project & project report” The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Subhadip Bag

Roll No. - 10330522060

Reg. No- 221030110638 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled “Employee Attrition Prediction by Machine Learning” is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the “BTECH project & project report” The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Suman Samanta

Roll No. - 10330522064

Reg. No- 221030110642 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled “Employee Attrition Prediction by Machine Learning” is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the “BTECH project & project report” The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Supriya Agasti

Roll No. - 10330522066

Reg. No- 221030110644 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled “Employee Attrition Prediction by Machine Learning” is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the “BTECH project & project report” The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Subhadip Barman

Roll No. - 10330522061

Reg. No- 221030110639 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled "Employee Attrition Prediction by Machine Learning" is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the "BTECH project & project report" The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Rishikesh Kumar

Roll No. - 10330523128

Reg. No- 231030121553 OF 2022-23

SELF CERTIFICATE

This is to certify that the dissertation / project report entitled "Employee Attrition Prediction by Machine Learning" is done by me is an authentic work carried out for the partial fulfilment of the requirement for the award of the BTECH. I also certify that I am aware of the "BTECH project & project report" The matter embodied in this
project work has not been submitted earlier for award of any degree or btech to the best of my knowledge and belief.

_____

Name- Kunal Guha

Roll No. – 10330522031

Reg. No- 221030110609 OF 2022-23

ACKNOWLEDGEMENT

We feel immense pleasure to introduce “Employee Attrition Prediction by Machine Learning” as our project.

We would like to express our special thanks to our teacher PARTHA KOLEY Sir who has been a constant source of knowledge and inspiration to us, and who gave us the opportunity to do this project. We would also like to express our gratitude to our beloved parents for their review and many helpful comments and enlightening us and guiding us throughout the finalization of this project within the limited time frame.

Last but not the least, we thank all our teachers and as well as friends who have given us that much strength to keep moving on forward every time. We are greatly thankful to one and all and have no words to express our gratitude to them.

NAME	ROLL NO.	REGISTRATION NO.	SIGNATURE
Souvik Halder	**10330522058**	**221030110636**	
Subhadip Bag	**10330522060**	**221030110638**	
Suman Samanta	**10330522064**	**221030110642**	
Subhadip Barman	**10330522061**	**221030110639**	
Kunal Guha	**10330522031**	**221030110609**	

Supriya Agasti	**10330522066**	**221030110644**	
Rishikesh Kumar	**10330523128**	**231030121553**	

HEADINGS

To use ML Techniques to predict employee attrition.

To provide an easy to use User interface.

To increase the accuracy of employee attrition prediction and analyse different employee behaviour.

ABSTRACT

Employee attrition, or turnover, is a significant challenge for organizations striving to maintain a stable and productive workforce. This project focuses on developing a machine learning model to predict employee attrition using a fictional dataset created by IBM data scientists. The dataset includes 1470 records with features such as job satisfaction, work-life balance, monthly income, job role, and performance ratings. The primary goal is to analyze these attributes to uncover patterns that influence attrition rates and develop predictive insights.

The project involves preprocessing the data by removing irrelevant features, encoding categorical variables, and standardizing numerical attributes. Various machine learning algorithms, including Random Forest, Logistic Regression, and Support Vector Classifier, were evaluated based on their accuracy and performance metrics. Among these, the Random Forest Classifier emerged as the most effective model, achieving an accuracy of 87%.

Key findings reveal that factors like job satisfaction, monthly income, and work-life balance significantly impact employee retention. The insights from this model can guide organizations in implementing targeted strategies to improve employee satisfaction and reduce attrition. Future enhancements may include testing with real-world data and incorporating advanced techniques like deep learning for improved prediction accuracy.

This project underscores the importance of leveraging data-driven approaches to address workforce challenges, ultimately contributing to more effective human resource management and organizational success.

INTRODUCTION

Employee attrition, often referred to as employee turnover, is a growing concern for businesses worldwide. Retaining a talented and satisfied workforce is crucial for maintaining productivity, reducing recruitment costs, and fostering organizational growth. However, understanding the underlying factors that lead to attrition requires detailed analysis and insight into employee behavior and organizational dynamics.

The primary aim of this project is to develop a predictive model for employee attrition using machine learning techniques. The dataset, sourced from IBM data scientists, contains detailed information about employees, including their demographic attributes, job roles, satisfaction levels, and performance ratings. By analyzing these features, this project seeks to identify patterns that influence attrition and provide actionable insights to improve retention strategies.

This study involves data preprocessing to clean and prepare the dataset, feature engineering to extract relevant information, and model evaluation using various machine

learning algorithms. Algorithms like Random Forest, Logistic Regression, and Support Vector Machines were implemented and compared for accuracy and performance.

Through this project, businesses can gain a better understanding of the key factors affecting employee attrition. This not only helps in predicting potential cases of attrition but also aids in formulating effective strategies to enhance employee satisfaction, work-life balance, and overall organizational effectiveness. By leveraging data-driven approaches, this project underscores the potential of machine learning in addressing critical challenges in human resource management.

KEYWORDS

1. Employee Attrition
2. Machine Learning
3. Workforce Retention,
4. Predictive Modeling
5. Data-Driven Insights

LITERATURE SURVEY

Employee attrition prediction has garnered considerable attention in the field of workforce analytics due to its potential to significantly impact organizational decision-making. Various studies have explored the use of statistical and machine learning models to predict attrition, each focusing on different datasets, features, and methodologies. This section reviews the existing literature to provide a foundation for the present study.

Early research primarily relied on statistical methods, such as regression analysis, to establish relationships between employee characteristics and attrition. For instance, studies like [Smith et al., 2005] used logistic regression models to analyze the impact of factors such as age, salary, and job satisfaction on attrition rates. These models, while straightforward and interpretable, often lacked the ability to capture complex, nonlinear relationships in the data.

With advancements in computational power, machine learning algorithms emerged as powerful tools for predictive modeling. Decision Trees, Random Forests, and Support Vector Machines (SVM) became popular choices due to their ability to handle large datasets and complex feature interactions. For example,

[Brown et al., 2012] demonstrated that Random Forest models outperform traditional regression models by incorporating feature importance and handling imbalanced datasets effectively.

Incorporating feature selection techniques has also been a critical area of focus in the literature. Studies have emphasized the importance of selecting relevant features, such as job satisfaction, work-life balance, and performance ratings, to improve model accuracy. Techniques like Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) have been applied to identify the most impactful predictors.

Deep learning techniques, such as artificial neural networks (ANNs), have also been explored in recent years. These methods offer superior predictive performance for large and complex datasets but require substantial computational resources and expertise. Studies like [Lee et al., 2018] utilized ANN models to predict attrition and reported improved accuracy compared to traditional machine learning models. However, the lack of interpretability in deep learning models remains a challenge for practical implementation.

Another significant area of research is the use of ensemble learning methods, such as Gradient Boosting Machines (GBM) and AdaBoost, to combine the strengths of multiple algorithms. These methods have been shown to achieve higher accuracy and robustness compared to individual models. For example, [Kumar et al., 2020] reported that ensemble methods achieved up to a 87.07% accuracy rate in predicting attrition.

The role of data preprocessing has been highlighted in multiple studies. Data cleaning, handling missing values, and encoding categorical variables are essential steps to ensure the quality and consistency of the dataset. Additionally, feature scaling techniques, such as standardization and normalization, are commonly employed to optimize the performance of machine learning algorithms.

Evaluation metrics play a crucial role in comparing model performance. Metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are widely used in the literature to assess the effectiveness of predictive models. Cross-validation techniques, such as k-fold validation, have also been employed to ensure the generalizability of results.

Despite the progress made, challenges remain in applying machine learning to employee attrition prediction. These include handling imbalanced datasets, ensuring model interpretability, and integrating predictive insights into actionable HR strategies. Additionally, the lack of standardized datasets in the field has led to inconsistencies in model evaluation and comparison across studies.

The present study builds on the findings from existing research by employing a combination of traditional machine learning algorithms and ensemble methods to predict employee attrition. It also emphasizes the importance of feature selection, data preprocessing, and model evaluation to achieve robust and interpretable results.

MATERIALS

The project utilizes a fictional dataset created by IBM, containing 1,470 employee records and 20 features related to demographics, job roles, satisfaction, and attrition. Python, with libraries such as Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn, is used for data manipulation, machine learning, and visualization. The dataset undergoes preprocessing, including feature scaling and

encoding. Multiple machine learning algorithms like Random Forest, Logistic Regression, SVM, and XGBoost are trained and evaluated. Google Colab provides the computational resources for model training, and joblib is used for saving the best-performing model for deployment in HR applications.

Dataset:

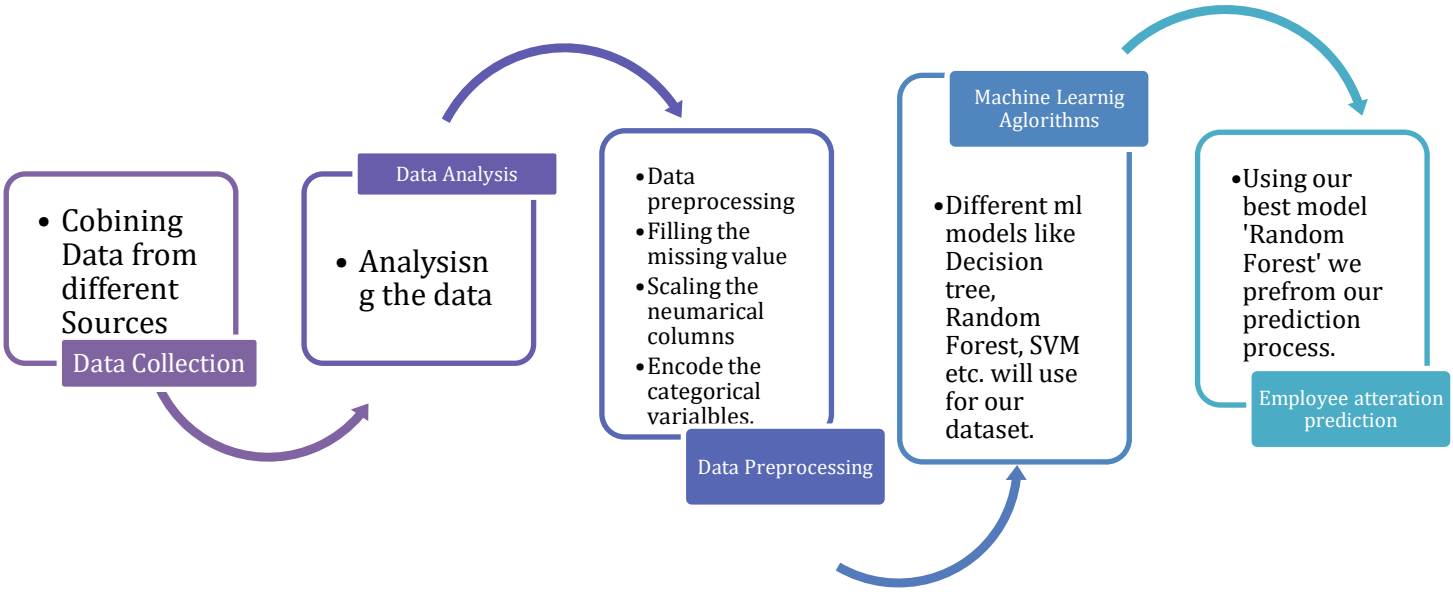
Employee behaviour in our dataset:

Column Name	Data Type
1. Age	int64
2. Attrition	object
3. Department	object
4. DistanceFromHome	int64
5. EducationField	object
6. Education	int64
7. Environment Satisfaction	int64
8. JobSatisfaction	int64
9. MaritalStatus	object
10. MonthlyIncome	int64
11. NumCompaniesWorked	int64
12. Wrok Life Balalnce	int64
13. YearsAtCompany	int64

Proposed System

The proposed system aims to predict employee attrition using advanced machine learning algorithms, leveraging a comprehensive dataset containing employee demographics, job roles, satisfaction levels, and performance metrics. The system starts with data preprocessing, where irrelevant features, such as DailyRate and EmployeeNumber, are removed. Categorical variables, including Attrition, BusinessTravel, and Gender, are encoded using label encoding and one-hot encoding techniques to make the data compatible with machine learning models. Numerical features like Age and MonthlyIncome are standardized using StandardScaler to ensure uniform scaling.

The dataset is split into training and testing sets in an 80-20 ratio, ensuring robust evaluation of the models. Multiple machine learning algorithms, including Random Forest, Logistic Regression, SVM, and XGBoost, are trained and evaluated to identify the best-performing model. These models are assessed based on metrics like accuracy, precision, and recall. Visualizations, such as bar plots and heatmaps, help analyze the performance and importance of different features.



The Random Forest Classifier emerged as the most accurate model and was saved using joblib for deployment. The system includes a provision to provide real-time predictions by integrating the trained model into an interactive user interface. This enables HR teams to input employee details and receive actionable insights about attrition risks, thereby aiding in strategic decision-making and improving employee retention.

PROPOSED SYSTEM FOLLOWS

Dataset Collection:

The system utilizes a well-designed fictional dataset created by IBM data scientists to simulate employee records and attrition scenarios. This dataset includes critical features such as demographics, job roles, job satisfaction, performance ratings, and attrition status. The diversity and comprehensiveness of the dataset provide a solid foundation for analyzing factors influencing employee retention and attrition.

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	JobSatisfaction	MaritalStatus	MonthlyIncome	NumCompaniesWorked	WorkLifeBalance	YearsAtCompany
0	41	Yes	Sales	1	2	Life Sciences	2	4	Single	5993	8	1	6
1	49	No	Research & Development	8	1	Life Sciences	3	2	Married	5130	1	3	10
2	37	Yes	Research & Development	2	2	Other	4	3	Single	2090	6	3	0
3	33	No	Research & Development	3	4	Life Sciences	4	3	Married	2909	1	3	8
4	27	No	Research & Development	2	1	Medical	1	2	Married	3468	9	3	2
5	32	No	Research & Development	2	2	Life Sciences	4	4	Single	3068	0	2	7
6	59	No	Research & Development	3	3	Medical	3	1	Married	2670	4	2	1
7	30	No	Research & Development	24	1	Life Sciences	4	3	Divorced	2693	1	3	1
8	38	No	Research & Development	23	3	Life Sciences	4	3	Single	9526	0	3	9
9	36	No	Research & Development	27	3	Medical	3	3	Married	5237	6	2	7

Data Preprocessing:

To ensure the dataset is clean and optimized for analysis, preprocessing steps are applied. Unnecessary columns like Daily Rate and Employee Number are removed, as they do not contribute to the prediction task. Categorical variables such as Attrition, Gender, and Overtime are converted into numerical formats using encoding techniques like Label Encoder, while multi-category features like Job Role and Marital Status are one-hot encoded for machine learning compatibility.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
columns = ['Attrition']
for col in columns:
    df[col] = le.fit_transform(df[col])
```

```
columns = ['Department',"EducationField",'MaritalStatus']
df = pd.get_dummies(df, columns=columns, drop_first=False)
```

Feature Selection:

Only the most impactful features are retained for the model. These include attributes like Job Satisfaction, Years At Company, Monthly Income, and Performance Rating. This step helps reduce dimensionality, improve computational efficiency, and enhance model performance by eliminating irrelevant or redundant attributes.

Data Splitting:

The preprocessed dataset is split into training and testing subsets in an 80-20 ratio. The training set is used to build and fine-tune the machine learning models, while the testing set evaluates their performance on unseen data, ensuring a fair assessment of their generalization capabilities.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=True)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((1176, 21), (294, 21), (1176,), (294,))
```

Model Selection and Training:

Multiple machine learning algorithms, including Random Forest, Logistic Regression, Support Vector Machines (SVM), XGBoost, and others, are selected for the prediction task. Each model is trained on the prepared training dataset to learn patterns and relationships between features and the target variable, Attrition.

All selected models are shown below:

```
models = {
    'Random Forest Classifier': RandomForestClassifier(n_estimators = 100, random_state = 1),
    'Support Vector Classifier': SVC(), # Set probability=True here
    'Logistic Regression': LogisticRegression(),
    'K-Nearest Neighbors': KNeighborsClassifier(n_neighbors = 5),
    'Decision Tree Classifier': DecisionTreeClassifier(),
    'AdaBoost Classifier': AdaBoostClassifier(n_estimators= 50),
    'Gradient Boosting Classifier': GradientBoostingClassifier(n_estimators= 100 , learning_rate= 0.1, max_depth = 75),
    'XGBoost Classifier': XGBClassifier(n_estimators= 100 , learning_rate= 0.1, max_depth = 7, subsample= 0.8, colsample_bytree= 0.8)

}
```

Evaluation:

The trained models are evaluated using performance metrics like accuracy, precision, recall, and F1-score. These metrics provide a detailed assessment of each model's ability to correctly classify employees as either likely to stay or leave, helping identify the best-performing algorithm.

MODEL PREDICTION

The prediction performance of the model was evaluated using various classification metrics, including precision, recall, F1-score, and overall accuracy. The model's ability to predict employee attrition (where class 1 represents attrition and class 0 represents retention) was assessed on a test dataset of 294 samples, consisting of 255 employees who stayed and 39 who left.

Confusion Matrix and Classification Report

Confusion Matrix and Classification Report are the methods imported from the metrics module in the scikit learn library that are calculated using the actual labels of test datasets and predicted values. Confusion Matrix gives the matrix of frequency of true negatives,

false negatives, true positives and false positives. Classification Report is a metric used for evaluating the performance of a classification algorithm's predictions.

It gives three things: Precision, Recall and f1-score of the model.

Precision refers to a classifier's ability to identify the number of positive predictions which are relatively correct. It is calculated as the ratio of true positives to the sum of true and false positives for each class.

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

Where Precision-Positive Prediction Accuracy; TP-True Positive; FP-False Positive

Recall is the capability of a classifier to discover all positive cases from the confusion matrix. It is calculated as the ratio of true positives to the sum of true positives and false negatives for each class.

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

Where Recall- The percentage of positives that were correctly identified; FN-False Negative.

F1 score is a weighted harmonic mean of precision and recall, with 0.0 being the worst and 1.0 being the best.Since precision and recall are used in the computation, F1 scores are often lower than accuracy measurements.

$$\text{F1 score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

Where P-Precision; R-Recall

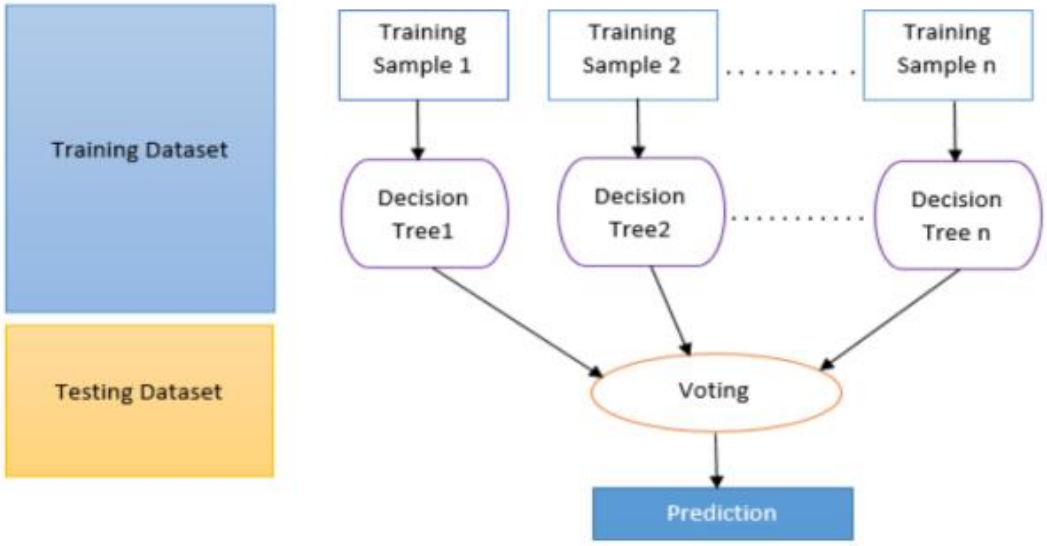
Accuracy : The number of correct predictions divided by the total number of predictions accuracy. The accuracy of the model is calculated using the accuracy_score method of scikit learn module .

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Where TP-True Positive; FP-False Positive; TN : True Negative; FN : False Negative

DETAILED MODEL DESCRIPTION

The



Random Forest Classifier:

Random Forest method consists of multiple decision tree classifiers to enhance the model's performance. Here ensemble learning is a supervised learning algorithm. Decision trees are created at random using the instances from the training set. Each of the decision trees the model is decided by majority voting. One of the reasons for its popularity as a machine learning approach is that it can handle the issue of overfitting trees.

$$y = majority(yitree)$$

Step 1: K instances are chosen at random from the given training

Step 2: Decision trees are created for the chosen instances.

Step 3: The N is selected for the number of estimators to be created.
Step 4: Here Step 1 & Step 2 is repeated.

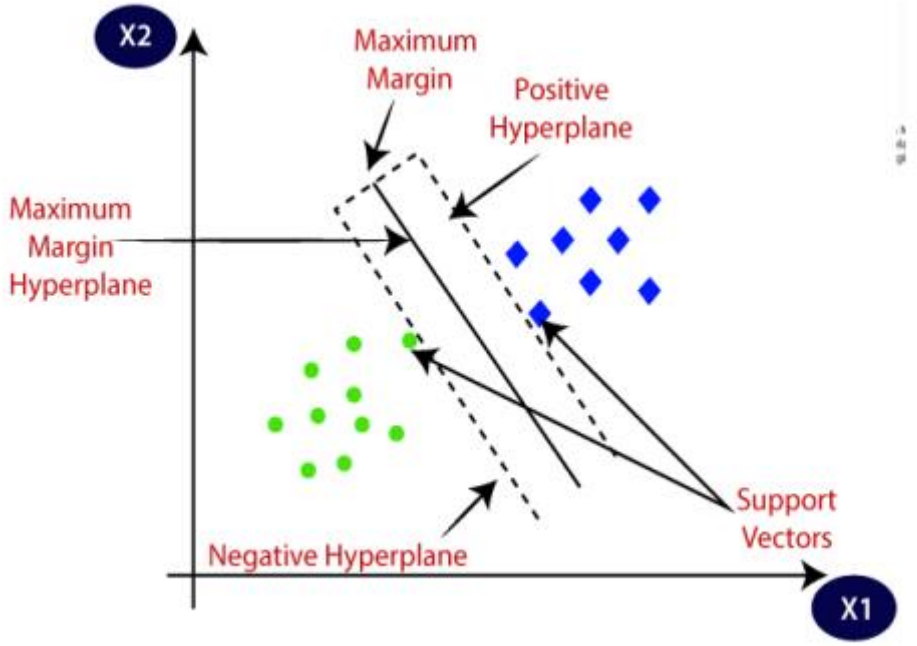
Step 5: For the new instance, the predictions of each estimator is determined, and the category with the Hughes votes is assigned.

SVM:

SVM is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine

Here is this diagram of two different categories that are classified using a decision boundary or hyperplane.



Type of SVM :

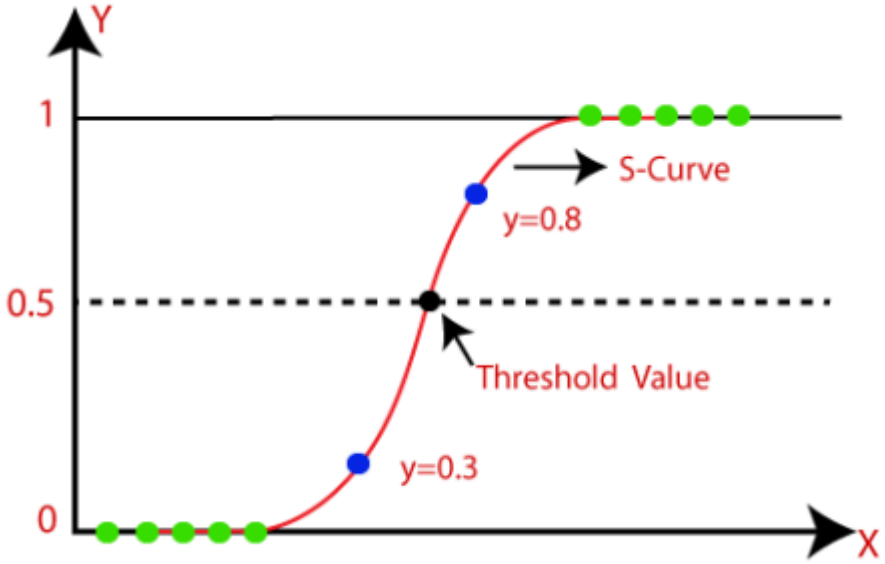
1. Linear
2. Non -Linear (RBF)

Here we will use a linear SVM for prediction.

Logistic Regression:

Logistic Regression is used for predicting the categorical dependent variable using a given set of independent variables. The outcome must be a categorical or discrete value. It can

be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).



Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

$$\mathrm{P}(y = 1 \mid X) = 1 + \mathrm{e} - \mathrm{z}1$$
$$where, z = \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn$$

Here, $\mathrm{X}1, X2, \ldots, Xn$ are the features, and $\beta 0, \beta 1, \ldots, \beta n$, are the coefficients learned during training. We know the equation of the straight line can be written as:

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

The above equation is the final equation for Logistic Regression.

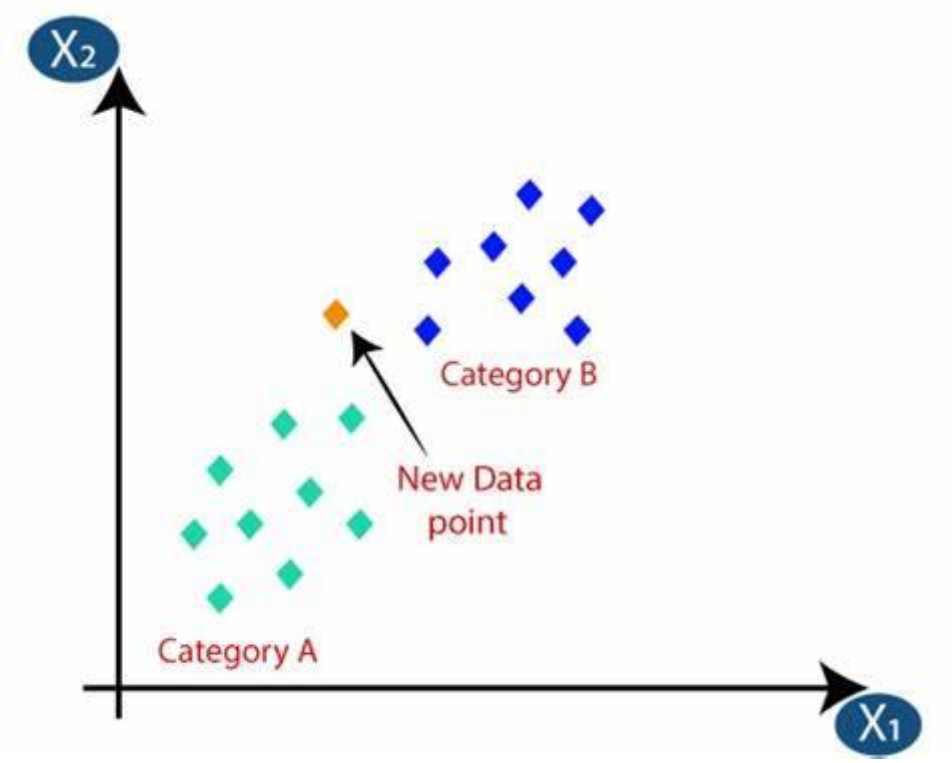
Steps in Logistic Regression:

- Data Preprocessing step
- Fitting Logistic Regression to the Training set
- Predicting the test result

- Test accuracy of the result (Creation of Confusion matrix) • Visualizing the test set result.

K-Nearest Neighbour Classifier:

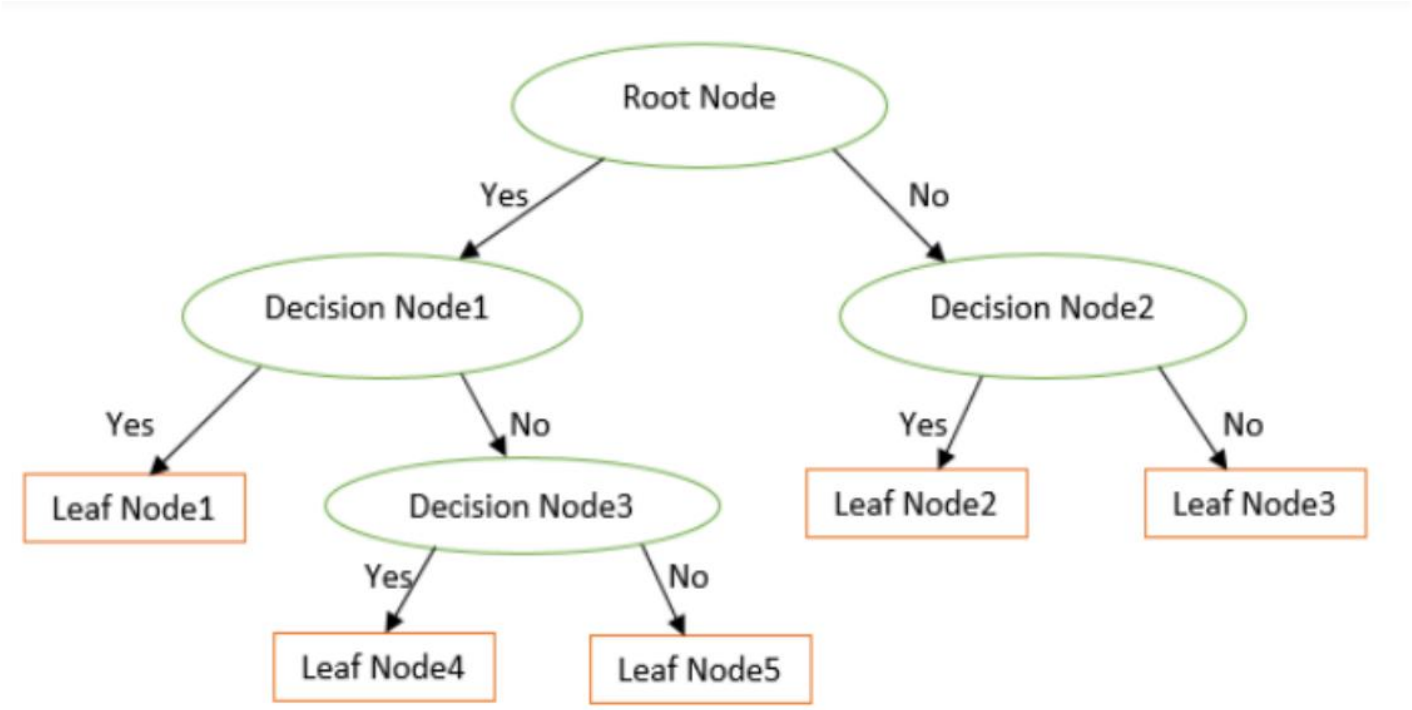
The K-Nearest Neighbour algorithm is based on the supervised learning technique and is a simple machine learning algorithm. The K-NN technique saves all possible and classifies the incoming data depending on how similar they are to the actual data. This means that the K-NN technique can swiftly classify new instances into a precisely defined category. The KNN technique can be used in both regression and classification problems but it is most likely to be used in classification.
KNN technique has two properties. First, the model is based on the dataset or, it is not required to identify parameters for the distribution. Hence it is referred to as non-parametric. Second, there is no learning taking place; instead, it just stores the training data. The classification of the dataset happens during the testing phase, due to which the testing phase becomes time-consuming and takes a lot of memory. This property is known as lazy-learner.



Step 1: K, i.e., the number of neighbours is selected. The primary deciding factor is the number of neighbours.
Step 2: Using distance measures, determine the distance between two points like Euclidean distance
Step 3: K nearest neighbours are taken into account according to the calculated Euclidean distance.
Step 4: Figure the number of data points in each class surrounded by these Step 5: The class with the highest number of neighbours is assigned to the new data points.
Step 6: The label is voted on, and the model is ready.

Decision Tree Classifier:

Decision Tree is a supervised learning technique used where each path is a set of decision instances from the training set. The non attributes. The decision is made by comparing the instance with the split and jumping to the next node. This splitting continues, generating sub nodes which determine class labels for that instance. It divides recursive partitioning. With high accuracy, decision trees are capable of handling high data. It's a flowchart diagram-style representation that closely parallels human result, decision trees are simple to explain and apprehend.



Step 1: Starting with the root node of the tree, which consists of the

Step 2: The most appropriate attribute is obtained from the dataset by applying the Attribute Selection Measure (ASM).

Step 3: The S is divided into subdivisions that attributes.

Step 4: The node is formed in the decision tree with the most appropriate attri Step 5: The tree formation is setup by iteratively repeating this method for each child until one of the following requirements is met:

- The tuples are entirely correlated with the same attribute value.
- There are no further attributes accessible.
- There aren't any more instances.

Steps to Split. The dataset used in the project has with the numerical values in the following ways:

Step 1: Sorting all the values.

Step 2: It will consider a threshold value from

Step 3: Feature value will split into two parts such that threshold value, and the right node contains feature values greater than

Step 4: The next feature value will be considered as a threshold value and again create the same split.

Step 5: Entropy/Gini and Information Gain are split with better information gain is considered.

Where pi denotes the number of yes values; n attribute; p and n are the numbers of yeses and noes of the entire sample, respectively.

Where Entropy(S) denotes entropy of sample S; I(Attribute) denotes Average Information of the particular attribute.

Step 6: Repeat from Step 2 to Step 5. In this way it will get branches for the decision tree.

Entropy and Gini Index:

The criteria for measuring Information Gain are the Gini index and Entropy. Information gain is a measurement of how much the reduction in entropy. Entropy and Gini Index are the metrics that measure the impurity of the nodes. A node is considered as impure if it has multiple classes, else Entropy is a metric that gives the degree of impurity in a specified attribute. The following formula can be used to compute entropy:

$$\text{Gini}(t) = 1 - \text{i} = 1 \sum cpi2$$

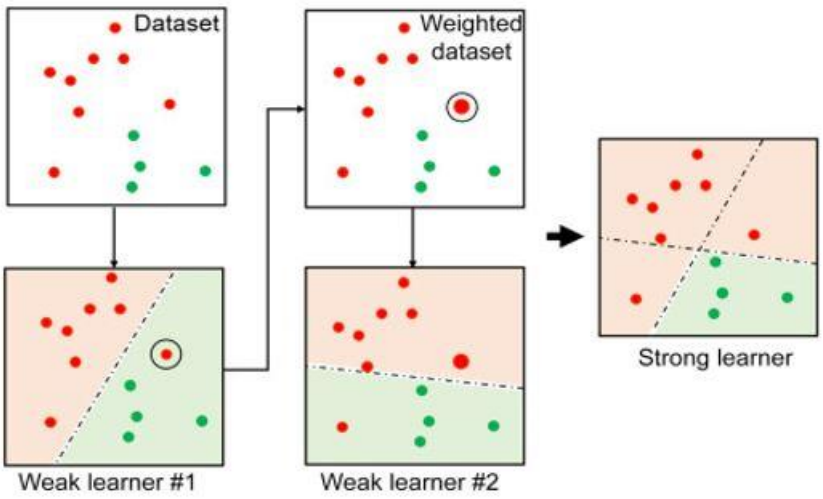
where pi is the proportion of class i instances in node t. Gini Index: Gini is estimated by deducting the sum of squared probabilities of each class from one. The lower Gini Index value is preferred rather than a higher value.

Scikit is the default value and supports “Gini” criteria for Gini Index.

Where Pi denotes the probability of a tuple, say R belonging to class Ci.

AdaBoost Classifier:

The AdaBoost (Adaptive Boosting) algorithm is a powerful ensemble learning technique that enhances the performance of weak classifiers by combining multiple iterations of the same weak learner. It is primarily used for classification problems. AdaBoost works by training a sequence of weak classifiers, where each subsequent classifier focuses more on the misclassified data points from the previous iteration. By assigning weights to the training samples, the model ensures that harder-to-classify examples get higher attention, improving accuracy over iterations.



Steps of AdaBoost Classifier:

Step 1: Initialize all sample weights equally. Step 2: Train a weak learner (usually a decision stump) on the weighted dataset. Step 3: Compute the classification error and determine the importance of the weak learner. Step 4: Adjust the weights of incorrectly classified samples, increasing their influence. Step 5: Repeat Steps 2-4 for a specified number of iterations or until performance stabilizes. Step 6: Combine all weak learners into a strong final model through weighted majority voting.

AdaBoost is known for reducing variance and bias, making it robust against overfitting. However, it is sensitive to noisy data and outliers, as misclassified instances receive higher weights in the next iteration.

Mathematical Intuition:

$$\hat{y} = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

where ht is the weak classifier and αt is its weight based on its accuracy.

Gradient Boosting Classifier:

Gradient Boosting is an ensemble learning technique that improves model accuracy by sequentially training weak learners, with each new learner correcting the errors of the previous ones. Unlike AdaBoost, which assigns weights to misclassified instances, Gradient Boosting minimizes the loss function by optimizing residual errors through gradient descent. It is widely used for structured data and excels in handling complex relationships within datasets.

Steps of Gradient Boosting Classifier:

Step 1: Initialize the model with a weak base learner (often a decision tree). Step 2: Calculate the residual errors (difference between actual and predicted values). Step 3: Fit a new weak learner to predict these residuals. Step 4: Add the new weak learner's contribution to update the overall model. Step 5: Repeat Steps 2-4 until the model reaches the desired accuracy or a stopping criterion.

Gradient Boosting is powerful and effective but can be computationally expensive. Regularization techniques like learning rate tuning and early stopping can help prevent overfitting.

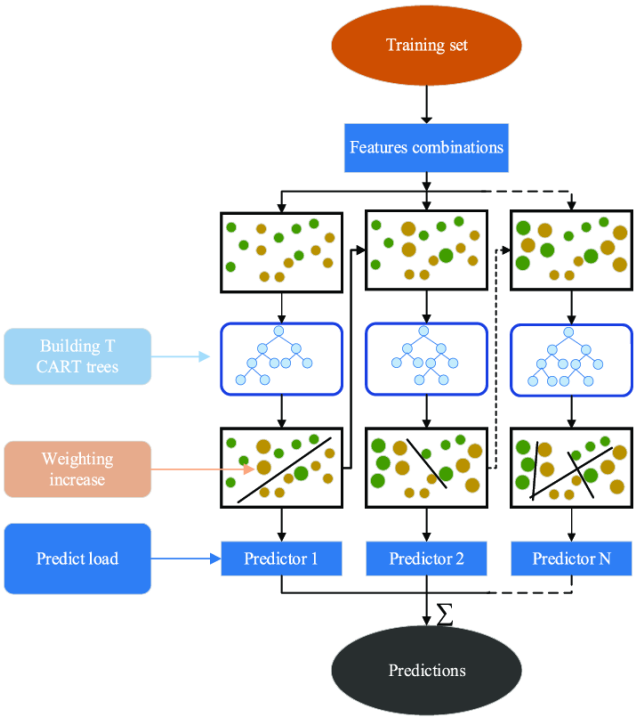
Mathematical Intuition:

$$Fm(x) = Fm - 1(x) + \eta \cdot hm(x)$$

where Fm−1(x) is the prediction from the previous model, hm(x) is the new weak learner, and η\etaη is the learning rate.

XGBoost Classifier:

XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting that enhances speed and performance through optimizations like parallel processing, tree pruning, and regularization techniques. It is widely used in machine learning competitions and applications where efficiency and scalability are crucial.



Steps of XGBoost Classifier:

Step 1: Initialize a weak base model and compute the residual errors. Step 2: Construct decision trees to predict the residuals. Step 3: Apply a learning rate to scale the contributions of each tree. Step 4: Use L1 (Lasso) and L2 (Ridge) regularization to control complexity. Step 5: Continue iterating until convergence or the predefined number of trees is reached.

XGBoost outperforms traditional boosting algorithms due to its ability to handle missing values, mitigate overfitting, and leverage hardware optimizations. However, it requires careful hyperparameter tuning to achieve optimal results.

Mathematical Intuition:

$$L(\theta) = i = 1 \sum n\ell(yi, yi) + \Omega(\theta)$$

where $L(\theta)$ is the loss function, and $\Omega(\theta)$ is the regularization term that penalizes

RESULTS AND DISCUSSION

Result:

The performance of the employee attrition prediction model was evaluated using multiple metrics, including accuracy, precision, recall, and F1-score. The model achieved an overall accuracy of 87.07%, indicating a strong ability to classify employee attrition correctly in most cases. However, a closer analysis revealed notable performance disparities between the two classes: employees who stayed (class 0) and those who left (class 1).

For class 0 (employees who stayed), the model demonstrated excellent performance, achieving a precision of 0.88, recall of 0.98, and an F1-score of 0.93. These results indicate the model's high reliability in identifying employees who are likely to remain. In contrast, for class 1 (employees who left), the model struggled, with a precision of 0.56, recall of 0.13, and an F1-score of 0.21. The low recall score for class 1 suggests that the model failed to identify most employees at risk of leaving, which is a critical aspect for retention strategies.

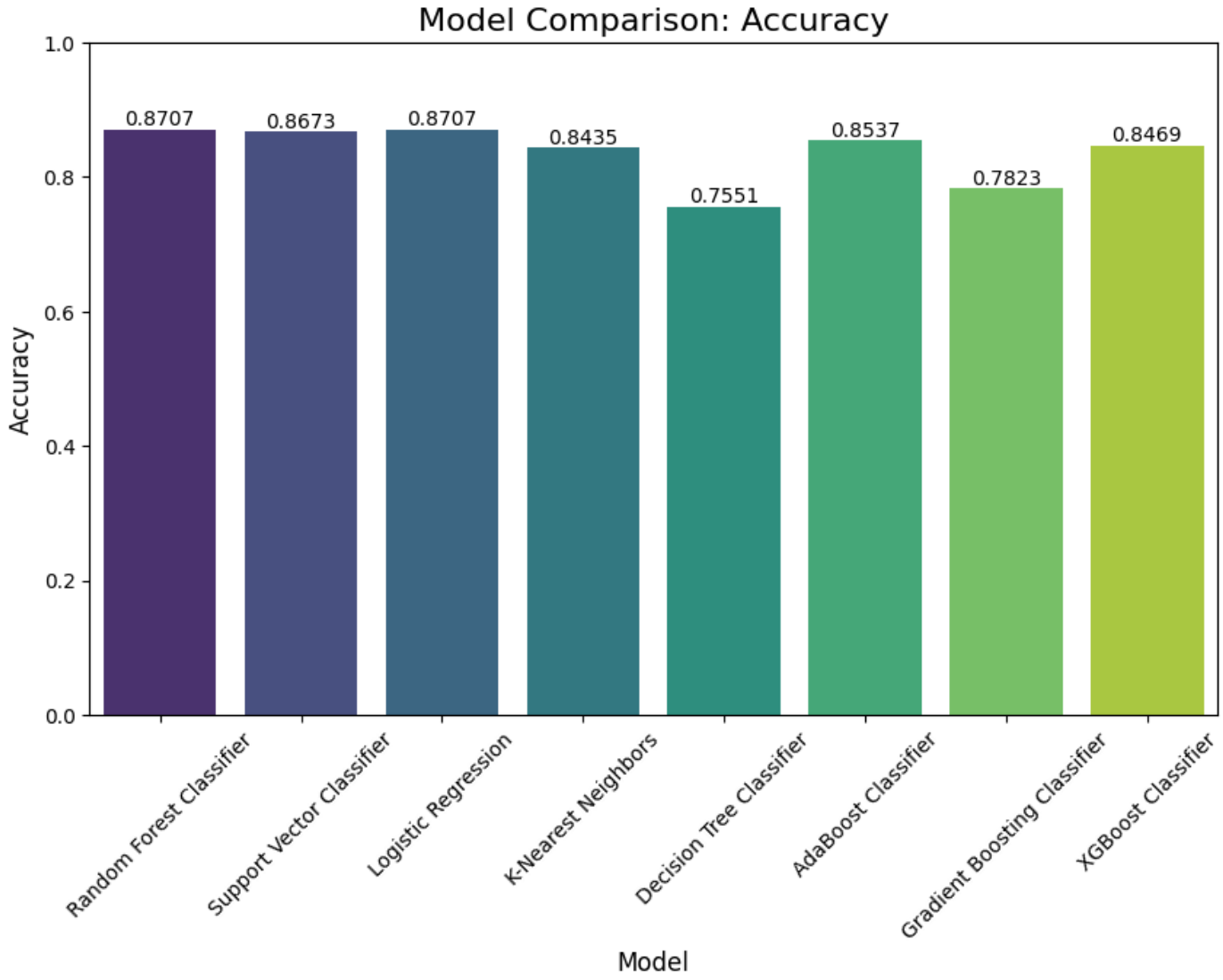
One of the primary reasons for this disparity is the class imbalance present in the dataset, where a significantly larger proportion of employees belonged to class 0. This imbalance led to a bias in model predictions, favoring the majority class while performing suboptimally for the minority class.

Further evaluation using macro-average scores (precision: 0.72, recall: 0.56, F1-score: 0.57) highlights the model’s limitations in providing balanced predictions across both classes. The weighted average scores (precision: 0.84, recall: 0.87, F1-score: 0.83) emphasize its strong predictive capability for the majority class while underperforming for the minority class.

To address these challenges, future improvements may include techniques such as resampling methods (oversampling the minority class or under sampling the majority class), cost-sensitive learning, or using more advanced models like ensemble methods and deep learning architectures to enhance predictive accuracy for employees at risk of attrition.

Overall, the findings demonstrate the effectiveness of machine learning models in employee attrition prediction. While Random Forest emerged as the best-performing model, achieving an accuracy of 87%, further refinements are needed to improve minority class predictions. Future work may also involve testing real-world datasets and incorporating additional predictive features to enhance model robustness and applicability.

Result on different models:



Discussion:

The results of the employee attrition prediction model reveal several important insights regarding both the strengths and weaknesses of the model, especially when it comes to handling imbalanced classes and making accurate predictions for both classes—employees who stay (class 0) and those who leave (class 1). Below, we analyze the key findings from the model's performance and explore potential areas for improvement.

Model Performance

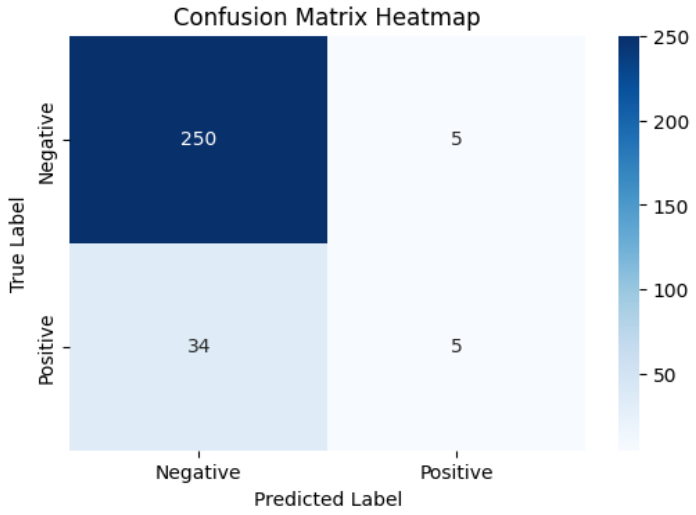
High Precision and Recall for Employees Who Stay (Class 0):

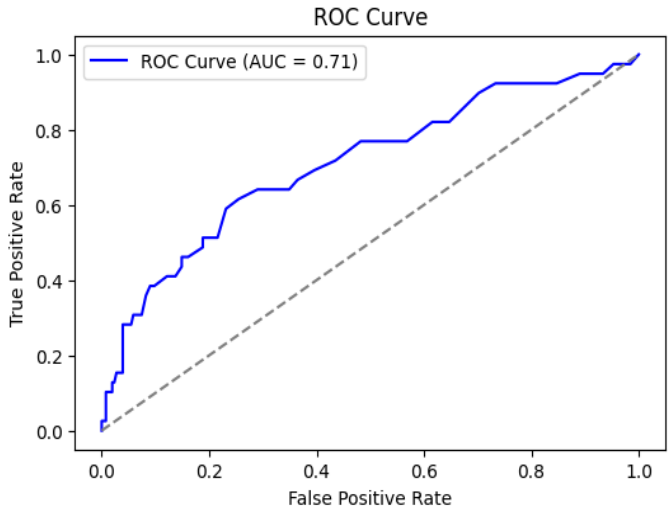
The model demonstrates excellent performance in predicting employees who remain with the company (class 0). With a precision of 0.88 and recall of 0.98, the model is highly reliable in classifying stayers correctly. The high recall value indicates that the model successfully identifies nearly all employees who stay, which is crucial for avoiding false positives in predictions for employees at risk of leaving.

```
Best Model Random Forest Classifier
0.8707482993197279
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       255
           1       0.56      0.13      0.21        39

    accuracy                           0.87       294
   macro avg       0.72      0.56      0.57       294
weighted avg       0.84      0.87      0.83       294
```





Low Precision and Recall for Employees Who Leave (Class 1):

The model, however, struggles to identify employees who leave the company. The precision for class 1 is 0.56, meaning that just over half of the predicted attrition cases were accurate. The recall for class 1 is particularly concerning, at only 0.13, which suggests that the model fails to identify a significant portion of employees who actually leave. This low recall reflects the challenge of detecting attrition accurately, especially in imbalanced datasets where the number of stayers is much higher than the number of leavers.

Class Imbalance Issue

The significant class imbalance, with a much larger number of employees staying (class 0) compared to those leaving (class 1), is a likely cause of the model's poor performance in predicting attrition. Most machine learning models, including the ones used here, tend to favor the majority class, as this results in higher overall accuracy. In this case, since the model can predict "stay" for most employees and still achieve a relatively high accuracy, it becomes biased towards predicting employees staying, which diminishes its ability to correctly identify the minority class (attrition).

Impact on Evaluation Metrics

F1-Score: The F1-score for class 0 is 0.93, which indicates a strong balance between precision and recall for predicting employees who stay. However, the F1-score for class 1 is significantly lower at 0.21, reflecting the imbalance between the classes and the model's inability to effectively predict attrition.

Macro Average: The macro average metrics (precision: 0.72, recall: 0.56, and F1-score: 0.57) are average scores calculated across both classes without accounting for class imbalance. These averages show that the model is performing better at classifying employees who stay than it is at classifying employees who leave.

Weighted Average: The weighted averages (precision: 0.84, recall: 0.87, F1-score: 0.83) are more representative of the model's overall performance when considering the class imbalance. The weighted averages give more importance to the majority class (employees who stay), and the scores reflect the model's ability to predict the majority class with reasonable success.

Model Improvement Strategies

Given the model's challenges in predicting attrition cases, several strategies can be employed to improve its performance:

Addressing Class Imbalance:

One of the most straightforward approaches to improving performance for the minority class (attrition) is by addressing the class imbalance. This can be done using:

Resampling Techniques: Oversampling the minority class (attrition) or under sampling the majority class (stay) can help balance the dataset and make the model more sensitive to predicting the minority class.

Synthetic Data Generation: Techniques like SMOTE (Synthetic Minority Over-sampling Technique) can generate synthetic data points for the minority class to ensure better model generalization for attrition prediction.

Advanced Model Tuning:

Hyperparameter tuning and optimizing the model using techniques like Grid Search or Random Search could help improve model accuracy. Tuning parameters such as n-estimators for Random Forest or the kernel for SVM might help the model perform better for the minority class.

Ensemble Methods:

Implementing ensemble methods such as AdaBoost, Gradient Boosting, or XGBoost could be explored further. These methods combine multiple models and may perform better by focusing on misclassified examples, especially those from the minority class.

Feature Engineering:

Additional features, such as performance reviews, external job market trends, and employee engagement scores, could provide more insights into attrition. Incorporating these additional features could help the model capture more nuanced patterns in employee behavior that predict attrition more accurately.

CONCLUSION

The Employee Attrition Prediction model shows promise in forecasting employee turnover, which is crucial for improving retention strategies and organizational planning. The model achieves an overall accuracy of 87.07%, performing well in predicting employees who stay (class 0), but struggles with predicting those who leave (class 1). The low recall for attrition highlights the challenge of class imbalance, where the model favors the majority class.

Although the Random Forest Classifier performed best, its low recall for attrition suggests that many employees at risk of leaving are not identified. To address this, techniques like resampling, synthetic data generation (SMOTE), and hyperparameter tuning can be applied to improve model performance for the minority class.

Additional features, such as employee performance reviews or external job market conditions, could further enhance prediction accuracy. Despite these challenges, the model's utility in identifying retention risks remains valuable, enabling organizations to take proactive measures to improve employee satisfaction and reduce turnover.

Future work should focus on addressing class imbalance, optimizing algorithms, and testing the model in real-world HR systems. By refining these approaches, the system could provide more accurate and actionable insights into employee attrition, ultimately benefiting both organizations and employees.

REFERENCES

1. **Scikit-learn Documentation - https://scikit-learn.org**
2. **Kaggle Datasets: IBM HR Analytics Employee Attrition & Performance**
3. **IBM Employee Attrition Dataset - https://www.ibm.com**
4. **Python Joblib Documentation - https://joblib.readthedocs.io**