TABLE OF CONTENT

ABSTRACT

Employee attrition, the departure of employees from an organization, poses a significant challenge for businesses. Proactively managing and predicting employee attrition is crucial for organizational stability and talent retention. This research explores the application of advanced unsupervised learning techniques—Isolation Forest and Gaussian Mixture Model (GMM)—to unravel patterns and factors contributing to employee attrition. Isolation Forest identifies anomalous behavior, while GMM provides insights into employee clusters. The study employs the Elbow Method, K-means Clustering with PCA, K-means Clustering with t-SNE, and LOF outliers with PCA to unveil patterns and clusters. These techniques aid in optimal cluster determination, segmentation insights, and anomaly detection. The holistic approach enhances interpretability and empowers organizations with nuanced understanding, fostering proactive retention strategies. Understanding employee attrition dynamics is vital for organizations aiming to develop effective strategies and ensure long-term organizational success.

KEYWORDS Employee Attrition, Unsupervised Learning, Prediction, Isolation Forest, Gaussian Mixture Model, K-means Clustering, Outliers, PCA

1.INTRODUCTION

Employee attrition, the phenomenon where employees voluntarily leave an organization, is a critical concern for businesses across industries. As the workplace landscape continues to evolve, organizations face the challenge of retaining skilled and experienced talent. Employee attrition can lead to a myriad of negative consequences, including loss of institutional knowledge, decreased team morale, and increased recruitment and training costs.

In this context, predicting and understanding employee attrition becomes imperative for organizational success. Proactive management strategies are essential to mitigate the adverse effects of attrition and foster a stable and productive work environment. Advanced predictive models, particularly those leveraging unsupervised learning techniques, offer a promising avenue for gaining insights into attrition patterns and contributing factors. This research focuses on the application of two advanced unsupervised learning techniques: Isolation Forest and Gaussian Mixture Model (GMM). Isolation Forest excels in identifying anomalous behavior within datasets, while GMM provides a nuanced understanding of clusters within the employee population. By integrating these models, we aim to present a comprehensive and holistic approach to employee attrition prediction.

The significance of this research lies in its potential to empower organizations with actionable insights, enabling them to implement targeted retention strategies. As we delve into the complexities of employee attrition dynamics, the utilization of advanced unsupervised learning techniques promises to contribute to the development of robust and effective retention strategies. By fostering a deeper understanding of attrition patterns, organizations can move from reactive to proactive talent management, ensuring long-term sustainability and success.

2.LITERATURE REVIEW

1. Employee Attrition: A Pervasive Challenge

Employee attrition, commonly referred to as employee turnover, is a pervasive challenge faced by organizations globally. Research in organizational psychology and management consistently emphasizes the multifaceted nature of attrition, acknowledging its impact on workforce stability and overall organizational performance. Attrition can occur for various reasons, including job dissatisfaction, lack of growth

opportunities, inadequate work-life balance, and organizational culture misalignment. Understanding the intricacies of why employees leave is fundamental to developing effective retention strategies.

2. Traditional Approaches to Attrition Prediction

Historically, organizations have relied on traditional approaches to predict employee attrition, often using statistical methods and basic machine learning techniques. While these methods provide insights, they may fall short in capturing the complexities of modern workplaces. The evolving nature of work, influenced by factors such as remote work, diverse organizational structures, and changing employee expectations, necessitates more sophisticated predictive models.

3. Unsupervised Learning in Employee Attrition Prediction

Recent research has witnessed a shift toward leveraging unsupervised learning techniques to enhance the accuracy and interpretability of attrition prediction models. Unsupervised learning allows the model to identify patterns and clusters within the data without relying on predefined labels. This is particularly advantageous in scenarios where the reasons for attrition may be diverse and not explicitly labeled in the dataset.

4. Isolation Forest: Identifying Anomalies

Isolation Forest, a novel unsupervised learning algorithm, has gained prominence in anomaly detection applications. By isolating instances in the data that are considered outliers, Isolation Forest excels in detecting anomalous behavior. In the context of employee attrition, this algorithm can pinpoint individuals or groups exhibiting behavior distinct from the norm, aiding in early identification of potential attrition risks.

5. Gaussian Mixture Model: Clustering Complexity

Gaussian Mixture Model (GMM) is a probabilistic model that assumes data is generated from a mixture of several Gaussian distributions. In employee attrition prediction, GMM can reveal underlying clusters within the workforce, providing a nuanced understanding of employee segments. This allows organizations to tailor retention strategies to specific employee groups, addressing their unique needs and concerns.

6. The Need for a Holistic Approach

As organizations strive to navigate the challenges posed by employee attrition, a holistic approach to prediction becomes essential. Integrating advanced unsupervised learning techniques, such as Isolation Forest and GMM, enables a more comprehensive understanding of attrition dynamics. By synthesizing anomaly detection and clustering, organizations can move beyond reactive responses and proactively shape their talent management strategies.

The literature underscores the evolving nature of employee attrition prediction, emphasizing the importance of leveraging advanced unsupervised learning techniques for a more nuanced and proactive approach.

3.STATEMENT OF THE PROBLEM

Employee attrition poses a substantial challenge for organizations, affecting their stability, productivity, and overall success. As the modern workforce landscape evolves, traditional approaches to predicting and mitigating attrition fall short in capturing the intricate patterns and diverse factors contributing to employee turnover. The problem at hand is the need for more advanced and nuanced predictive models that can comprehensively address the complexities of employee attrition in contemporary organizational settings.

4. SCOPE OF STUDY

The scope of this study is to investigate how an unsupervised machine learning model can be applied to large Fictional dataset on HR Employee attrition and performance to be able to predict employee attrition and to investigate practitioners' openness to use such models for decision making within an organization. That is, this study will aim to analyze the possibility to predict employee attrition based on personnel data set on HR employee attrition and performance and what effects practitioners think that being able to predict employee attrition could have on organizational performance.

5.OBJECTIVE OF STUDY:

The objective of this study is to analyze how objective factors influence employee attrition, in order to identify the main causes that contribute to a worker's decision to leave a company, and to predict whether a particular employee will leave the company. The study aims to employ unsupervised learning for employee attrition prediction, unveiling hidden patterns and clustering employees based on shared characteristics. This study will explore interpretability, ethical considerations, and the generalizability of results, providing a comprehensive framework for organizational implementation.

6.METHODOLOGY:

i. Data Sources:

The dataset is taken form Kaggle, it is a data science competition platform and online community of data scientists and machine learning practitioners under Google LLC.

Dataset link - https://www.kaggle.com/datasets/patelprashant/employee-attrition

ii. Data Preprocessing:

Data preprocessing includes 5 steps. They are Importing Libraries and Reading the dataset, check the missing values and considering the co relational heatmap, separating the independent and dependent variables, converting the data into numpy array and performing encoding on categorical variables and splitting the dataset for training and testing.

Firstly, we need to import the data to the operating environment. For loading the data sets and to preprocess them we need to import the libraries such as pandas, numpy and matplotlib for visualization. Then check for any missing values in the dataset. Separate the independent and dependent variables. Now perform encoding on categorical variables. Encoding categorical variables is that we are not able to pass the text data of the system, since these algorithms include the mathematical calculations that do not support the commands, so they have to be coded in a numerical variable. And we will encrypt it using a binary representation, and without the assignment of the numbers straight, because the assignment can be related to their priority. For the coding of categorical, text variables, we have used the sklearn package. This can be done by converting the data to a numpy array. We have splitted the dataset as 70% of the data to train the model and 30% to test the model. For this we need to import the "train_test_split" from sklearn package. In this splitting we use class which have attributes like test_size that specifies the percentage of test data and random_state that can have values 0 or 1 which is used to set the test data from the dataset. In this the variables x_train, y_train refers to the training data and x_test, y_test refers to the test data. The train values are passed for training the model and the test values are passed for testing the model developed.

iii. Description of the tools used:

1. K – means clustering:

k-means clustering categorizes employees into distinct groups based on shared characteristics, allowing organizations to identify patterns and potential risks of attrition where each cluster represents a group of employees with similar attributes such as job satisfaction, performance, and engagement.

2. Dimentionality reduction using t-SNE:

t-SNE focuses on preserving local similarities in high-dimensional data, making it particularly useful for revealing intricate structures within workforce attributes. By projecting multidimensional employee data onto a lower-dimensional space, t-SNE allows for the identification of clusters or groups with similar characteristics.

3. Outlier Detection Using Local Outlier Factor:

Outlier Detection using Local Outlier Factor (LOF) is for identifying individuals who exhibit behaviors significantly different from their peers within the workforce. LOF assesses the local density of each employee's attributes and highlights those with substantially lower densities as potential outliers.

4. Isolation Forest:

Isolation Forest excels at isolating and identifying unusual patterns or outliers within a dataset. When applied to employee attrition, the algorithm can effectively isolate individuals who exhibit behaviors or characteristics that deviate significantly from the norm, indicating a higher likelihood of leaving the organization.

5. One Hot Encoding:

One-Hot Encoding is a used to represent categorical variables, such as job roles or departments, as binary vectors. Each category is assigned a unique binary value, and the presence or absence of a category is denoted by 1 or 0, respectively.

6. Gaussian Mixture Model:

Gaussian Mixture Model (GMM) assumes the data is generated by a mixture of several Gaussian distributions. Each Gaussian component represents a distinct cluster or group of employees with shared characteristics related to attrition risk. GMM can be applied to identify underlying patterns in the workforce by capturing the inherent variability in employee attributes. By modeling the data distribution as a mixture of Gaussian components, GMM is capable of accommodating complex structures within the dataset, allowing for a nuanced understanding of different employee segments.

7.ANALYSIS

i. Algorithms used:

The employee attrition analysis employed a variety of algorithms to gain insights into the factors influencing employee turnover. These algorithms include:

- K-means clustering:
  This algorithm segments the employee data into distinct clusters based on their similarities, allowing for the identification of patterns and trends among different employee groups.
- t-SNE (t-distributed Stochastic Neighbor Embedding):

This dimensionality reduction technique projects high-dimensional employee data into a lower-dimensional space, facilitating visualization and analysis of complex relationships between variables.

- Local Outlier Factor (LOF):
  This algorithm detects outliers in the dataset, identifying individuals whose characteristics deviate significantly from the majority of the population.

- Isolation Forest:
  This outlier detection algorithm isolates anomalies by iteratively partitioning the data until each instance is either marked as an outlier or considered part of the inlier group.

- Gaussian Mixture Model (GMM):
  This probabilistic model assumes that the data is generated from a mixture of Gaussian distributions, enabling the identification of hidden patterns and underlying clusters

ii. ML Tools and techniques

The analysis utilized a range of machine learning tools and techniques to effectively analyze and interpret the employee attrition data. These tools include:

- Exploratory Data Analysis (EDA):
  EDA involved examining the data's distribution, patterns, and relationships between variables to gain a comprehensive understanding of its characteristics.

- Dimensionality reduction techniques (PCA, t-SNE):
  These techniques were employed to reduce the dimensionality of the data, making it more manageable for visualization and analysis.

- Outlier detection algorithms (LOF, Isolation Forest):
  These algorithms were used to identify outliers in the data, which could potentially represent exceptional cases or data errors.

- Clustering algorithms (K-means, GMM):
  These algorithms were applied to group similar employees together, revealing underlying patterns and hidden segments within the data.

iii. Exploratory Data Analysis:

The EDA phase involved a thorough examination of the employee data to uncover its characteristics and potential insights. This included:

- Visualizing the distribution of numerical variables:
  Histograms, box plots, and density plots were used to visualize the distribution of numerical features, providing insights into their central tendency, spread, and potential outliers.

- Data cleaning:
  Missing values were handled appropriately, and outliers were identified and considered for removal.

- Examining the relationships between variables:

Scatter plots and correlation matrices were employed to explore the relationships between different variables, identifying correlations and patterns.

- Analyzing categorical variables:
  Frequency counts and bar charts were used to understand the distribution of categorical variables, such as job roles and departments.

iv. Code and Output

The analysis was implemented using Python and various machine learning libraries, including scikit-learn and matplotlib. The code produced outputs such as:

- Visualizations of employee clusters:
  Scatter plots and cluster labels were used to visualize the results of K-means and GMM clustering algorithms, revealing distinct groups of employees.

- Outlier detection plots:
  Scatter plots with color-coded outlier labels were used to identify outliers detected by LOF and Isolation Forest algorithms.

- Performance metrics for outlier detection models:
  Precision, recall, and F1-score were calculated to evaluate the performance of outlier detection models in identifying true outliers.

- Prediction results for employee attrition:
  The Isolation Forest model was trained and utilized to predict whether a specific employee is likely to leave the company based on provided input features.

CODE:

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from sklearn.neighbors import LocalOutlierFactor
from sklearn.mixture import GaussianMixture
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import IsolationForest
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
import numpy as np
from sklearn.mixture import GaussianMixture
import plotly.graph_objects as go
import streamlit as st
```

```
# Load the CSV file into a DataFrame
file_path = "D:\\DCS\\SEM- 5\\5. HR Lab\\Project\\daatset.csv"
df = pd.read_csv(file_path)

# Check for null values in the DataFrame
null_values = df.isnull().sum()
print("Count of null values in each column:")
print(null_values)

# Handling missing values (as shown in the previous code)
# Streamlit code
st.title("Employee Attrition with Unsupervised Learning")
st.sidebar.title("Employee Attrition Model")
visualization_option = st.sidebar.selectbox("Select", ["Elbow Method Plot", "K-means Clusters with
PCA", "K-means Clusters with t-SNE", "LOF Outliers with PCA",
                                        "Isolation Forest Outliers", "Gaussian Mixture Model
Clusters","Prediction"])
# Descriptions for the selected visualization
descriptions = {
   "Elbow Method Plot": "Determine the optimal number of clusters using within-cluster sum of
squares.",
   "K-means Clusters with PCA": "Visualizes K-means clustering results in two-dimensional space
using Principal Component Analysis (PCA) for dimensionality reduction.",
   "K-means Clusters with t-SNE": "Displays K-means clustering results in two-dimensional space
using t-distributed Stochastic Neighbor Embedding (t-SNE) for dimensionality reduction.",
   "LOF Outliers with PCA": "Identifies and visualizes outliers in the data using Local Outlier Factor
(LOF) algorithm and presents the results in two-dimensional space using Principal Component
Analysis (PCA).",
   "Isolation Forest Outliers": "Detects outliers in the data using the Isolation Forest model",
   "Gaussian Mixture Model Clusters": "Applies Gaussian Mixture Model (GMM) clustering to the
data and visualizes the resulting clusters, specifically focusing on certain selected features like age,
job level, and monthly income.",
   "Prediction" :"The Isolation Forest model is trained and utilized for predicting whether a specific
employee is likely to leave the company based on the provided input features."
}
# Display the selected visualization description in the sidebar
st.sidebar.subheader(descriptions.get(visualization_option, ""))

# Select columns for clustering
columns_for_clustering = ['Age', 'MonthlyIncome', 'JobLevel']  # Add more columns as needed

# Select data for clustering
data_for_clustering = df[columns_for_clustering]

# Perform feature scaling if required (optional but recommended for K-means)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
scaled_data = scaler.fit_transform(data_for_clustering)
# Choosing the number of clusters using the elbow method
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    inertia.append(kmeans.inertia_)

# Performing K-means clustering
if visualization_option == "Elbow Method Plot":
    st.subheader("Elbow Method Plot")
    plt.figure(figsize=(8, 6))
    plt.plot(range(1, 11), inertia, marker='o', linestyle='--')
    plt.xlabel('Number of clusters')
    plt.ylabel('Inertia')
    plt.title('Elbow Method')
    st.pyplot(plt)


# Choosing the optimal number of clusters
optimal_clusters = 3  # Change this according to your elbow plot

# Performing K-means clustering
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
kmeans.fit(scaled_data)


# Adding cluster labels to the DataFrame
df['Cluster_KMeans'] = kmeans.labels_

# Visualizing K-means clusters
# Dimensionality reduction using PCA
pca = PCA(n_components=2)
data_pca = pca.fit_transform(scaled_data)

# Visualizing K-means clusters with PCA
if visualization_option == "K-means Clusters with PCA":
    st.subheader("K-means Clusters with PCA")
    # code to create the scatter plot with K-means clusters and PCA
    plt.scatter(data_pca[:, 0], data_pca[:, 1], c=df['Cluster_KMeans'], cmap='viridis')
    plt.xlabel('Principal Component 1')
    plt.ylabel('Principal Component 2')
    plt.title('PCA Components with K-means Clusters')
    plt.show()
    st.pyplot(plt)

# Dimensionality reduction using t-SNE
tsne = TSNE(n_components=2)
```

```
data_tsne = tsne.fit_transform(scaled_data)

# Visualizing t-SNE components with K-means clusters
if visualization_option == "K-means Clusters with t-SNE":
        st.subheader("K-means Clusters with t-SNE")
        # code to create the scatter plot with K-means clusters and t-SNE
        plt.scatter(data_tsne[:, 0], data_tsne[:, 1], c=df['Cluster_KMeans'], cmap='viridis')
        plt.xlabel('t-SNE Component 1')
        plt.ylabel('t-SNE Component 2')
        plt.title('t-SNE Components with K-means Clusters')
        plt.show()
        st.pyplot(plt)


# Novelty and Outlier Detection using Local Outlier Factor (LOF)
lof = LocalOutlierFactor(n_neighbors=20, contamination=0.1)
outlier_labels = lof.fit_predict(scaled_data)

# Visualizing LOF Outliers
if visualization_option == "LOF Outliers with PCA":
    st.subheader("LOF Outliers with PCA")
    # Your code to create the scatter plot with LOF outliers and PCA
    plt.scatter(data_pca[:, 0], data_pca[:, 1], c=outlier_labels, cmap='viridis')
    plt.xlabel('Principal Component 1')
    plt.ylabel('Principal Component 2')
    plt.title('LOF Outliers')
    plt.show()
    st.pyplot(plt)

# Splitting X and Y
# X contains the features, and y contains the target variable
# Split the data into training and testing sets
# And X_test and y_test for evaluating its performance
X = df[['Age','BusinessTravel','DailyRate','Department','DistanceFromHome',
'Education','EducationField','EmployeeCount','EnvironmentSatisfaction','HourlyRate','JobInvolveme
nt','JobLevel'
,'JobRole','JobSatisfaction','MaritalStatus','MonthlyIncome','MonthlyRate','NumCompaniesWorked','
OverTime','PercentSalaryHike','PerformanceRating','RelationshipSatisfaction','StandardHours','Stock
OptionLevel','TotalWorkingYears','TrainingTimesLastYear','WorkLifeBalance','YearsAtCompany','Ye
arsInCurrentRole','YearsSinceLastPromotion','YearsWithCurrManager']]
Y = df[['Attrition', 'EmployeeNumber', 'Over18','Gender']]  # 'Attrition' is the target variable
print(X)
print(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
print(X_train)
print(X_test)
```

```
# Outlier Analysis
# Isolation Forest
# Assuming 'df' is your DataFrame
# Identify categorical columns
categorical_columns = ['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole',
'MaritalStatus', 'Over18', 'OverTime']
# Identify numerical columns
numerical_columns = ['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount',
'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement', 'JobLevel',
'JobSatisfaction', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike',
'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany',
'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager']
# Split data into features (X) and target variable (y)
x = df[numerical_columns+categorical_columns]
y = df['Attrition']  # Assuming 'Attrition' is your target variable

# Create a ColumnTransformer to apply one-hot encoding to categorical columns
preprocessor = ColumnTransformer(
    transformers=[
        ('categorical', OneHotEncoder(), categorical_columns)
    ],
    remainder='passthrough'  # Pass through the non-categorical columns
)



# Create a pipeline with preprocessing and Isolation Forest
clf = Pipeline([
    ('preprocessor', preprocessor),
    ('isolation_forest', IsolationForest(contamination=0.1, random_state=42))  # Adjust contamination
based on your data
])

# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
# Print information about x_train
# Print column names in x_train
print("Column names in x_train:", x_train.columns)

# Fit the pipeline on the training data
clf.fit(x_train)

# Predict on the test set
y_pred = clf.predict(x_test)
print(y_pred)
y_true_numeric = np.where(y_test == 'No', 1, -1)

# Visualize Isolation Forest Outliers without PCA
```

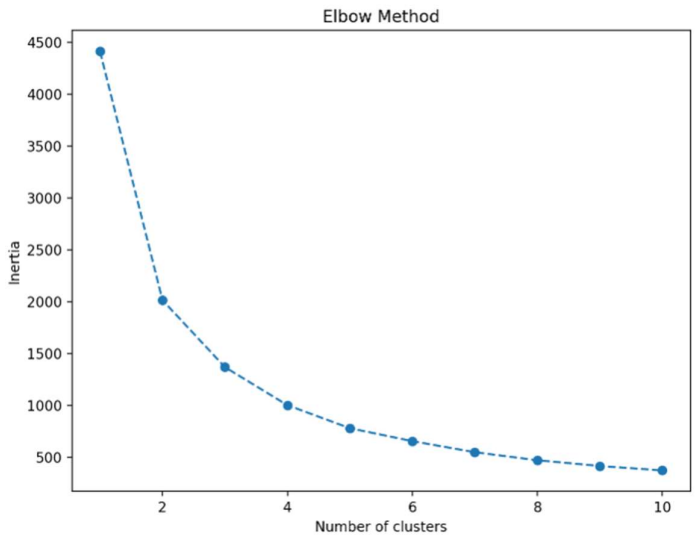
```
if visualization_option == "Isolation Forest Outliers":
    st.subheader("Isolation Forest Outliers")
    fig, ax = plt.subplots()
    # Choose two numerical features for the plot, e.g., 'Age' and 'MonthlyIncome'
    feature1 = 'Age'
    feature2 = 'MonthlyIncome'
    # Scatter plot
    scatter = ax.scatter(x_test[feature1], x_test[feature2], c=y_pred, cmap='viridis')
    # Legend
    legend = ax.legend(*scatter.legend_elements(), title="Outliers")
    ax.add_artist(legend)
    # Axis labels and title
    ax.set_xlabel(feature1)
    ax.set_ylabel(feature2)
    ax.set_title('Isolation Forest Outliers')
    st.pyplot(fig)
    # Print classification report
    st.subheader("Classification Report")
    classification_rep = classification_report(y_true_numeric, y_pred, output_dict=True)
    st.table(classification_rep)

if visualization_option == "Prediction":
    st.subheader("Prediction Model")
    # Selecting only the relevant columns
    selected_columns = ['Age', 'MonthlyIncome', 'JobLevel', 'JobSatisfaction']

    # Creating a new DataFrame with only the selected columns
    selected_data = df[selected_columns]

    # Split data into features (X) and target variable (y)
    X = selected_data
    y = df['Attrition']  # Assuming 'Attrition' is your target variable

    # Create a ColumnTransformer to apply one-hot encoding to categorical columns
    preprocessor = ColumnTransformer(
        transformers=[
            ('categorical', OneHotEncoder(), [])  # No categorical columns in this case
        ],
        remainder='passthrough'  # Pass through the non-categorical columns
    )

    # Create a pipeline with preprocessing and Isolation Forest
    clf = Pipeline([
        ('preprocessor', preprocessor),
        ('isolation_forest', IsolationForest(contamination=0.1, random_state=42))  # Adjust
contamination based on your data
    ])
    # Split the data into training and testing sets
```

```
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Fit the pipeline on the training data
    clf.fit(X_train)

    # Predict on the test set
    y_pred = clf.predict(X_test)
    y_true_numeric = np.where(y_test == 'No', 1, -1)

   # Collect user input
    user_age_input = st.text_input("Enter Age", 25)
    user_monthly_income_input = st.text_input("Enter Monthly Income", 5000)
    user_job_level_input = st.selectbox("Select Job Level", [1, 2, 3, 4, 5], index=2)
    user_job_satisfaction_input = st.selectbox("Select Job Satisfaction", [1, 2, 3, 4, 5], index=3)

    # Create a DataFrame with user input
    user_data = pd.DataFrame({
        'Age': [user_age_input],
        'MonthlyIncome': [user_monthly_income_input],
        'JobLevel': [user_job_level_input],
        'JobSatisfaction': [user_job_satisfaction_input],
    })

    # Make predictions
    user_pred = clf.predict(user_data)

    # Display prediction result
    st.subheader("Prediction Result")
    if user_pred[0] == -1:
        st.error("The model predicts that the employee is likely to leave the company.")
    else:
        st.success("The model predicts that the employee is likely to stay with the company.")


# Gaussian Mixture model
selected_columns = ['Age', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'MonthlyRate',
                    'NumCompaniesWorked', 'PercentSalaryHike', 'PerformanceRating', 'WorkLifeBalance']

selected_data = df[selected_columns]
def fit_and_plot_gmm(selected_columns, n_components=4):
    # Fit Gaussian Mixture Model
    gmm = GaussianMixture(n_components=n_components)
    gmm.fit(selected_columns)

    # Predict cluster labels
    labels = gmm.predict(selected_columns)

    # Add cluster labels to the DataFrame
```

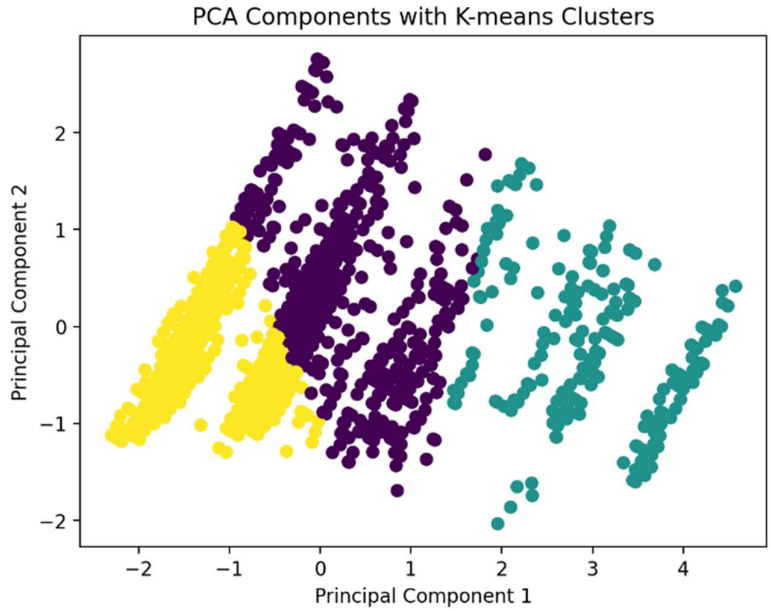
```
            selected_columns['cluster'] = labels

            # Plot clusters
            plot_clusters(selected_columns)

        def plot_clusters(selected_columns):
            colors = ['blue', 'green', 'cyan', 'black']

            # Scatter plot for each cluster
            for k in range(selected_columns['cluster'].nunique()):
                cluster_data = selected_columns[selected_columns['cluster'] == k]
                plt.scatter(cluster_data['Age'], cluster_data['MonthlyIncome'], c=colors[k], label=f'Cluster {k}')
        print("Before GMM:", selected_data.shape)
        fit_and_plot_gmm(selected_data)
        print("After GMM:", selected_data.shape)

        # Fit and plot GMM with the selected features
        # Add labels and title
        if visualization_option == "Gaussian Mixture Model Clusters":
            st.subheader("Gaussian Mixture Model Clusters")
            plt.xlabel('Age')
            plt.ylabel('MonthlyIncome')
            plt.title('Gaussian Mixture Model Clustering')
            plt.legend()
            plt.show()
    st.pyplot(plt)
```

OUTPUT:

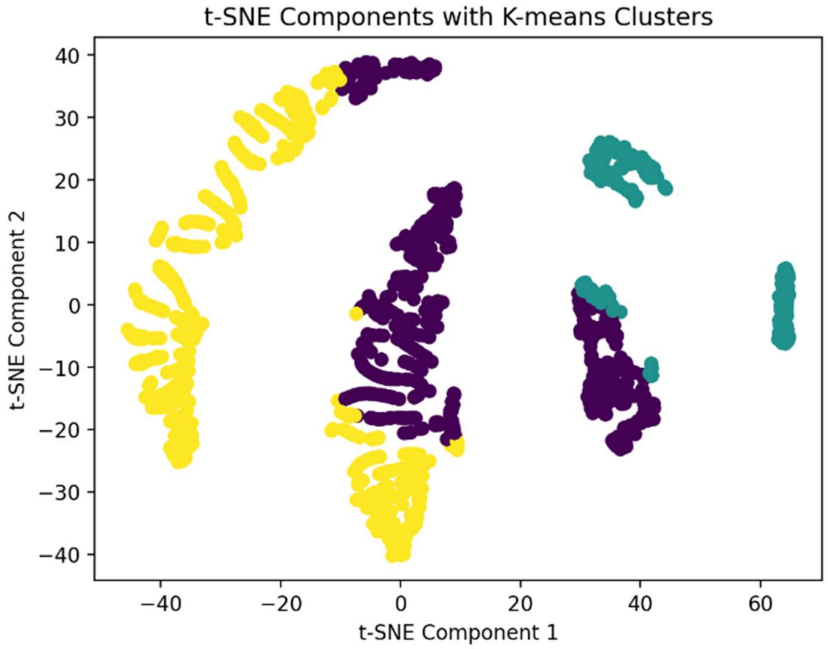
Elbow Method Plot



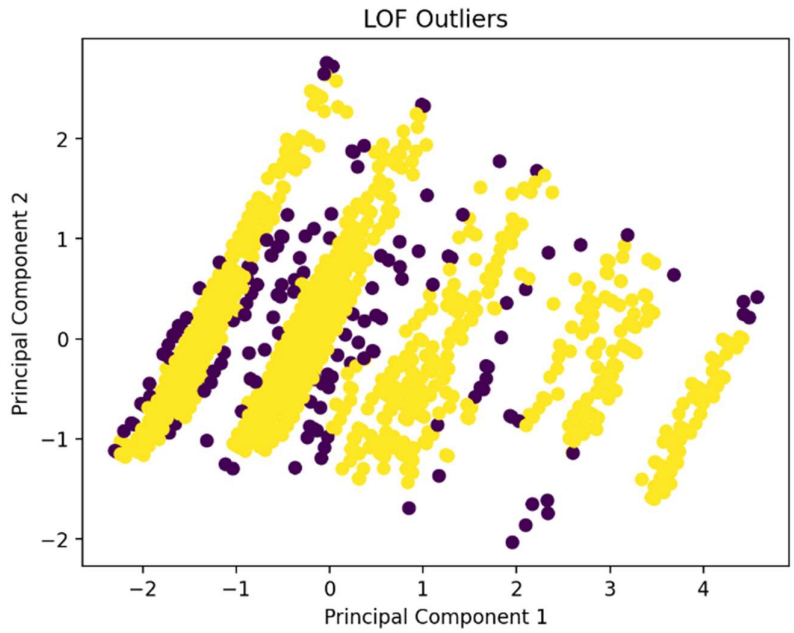
K-means Clusters with PCA



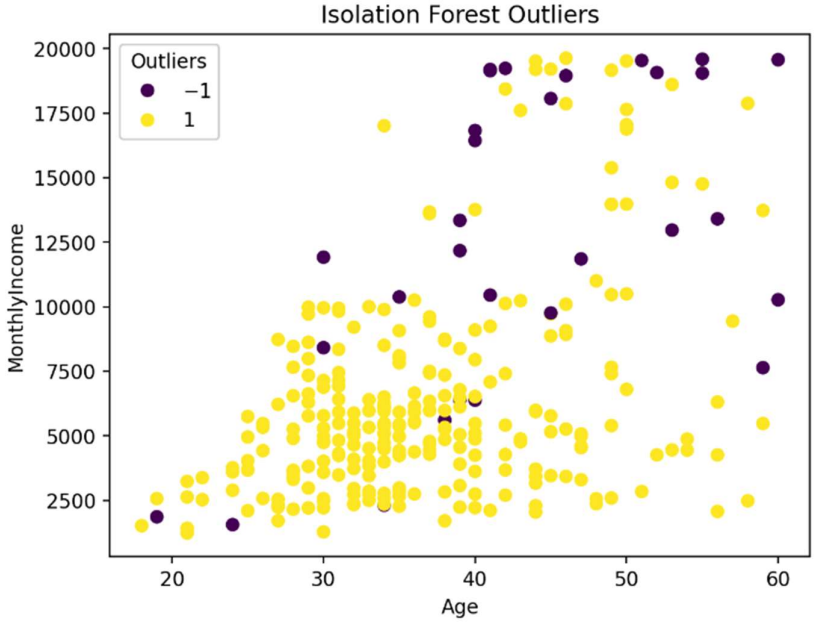
K-means Clusters with t-SNE



LOF Outliers with PCA



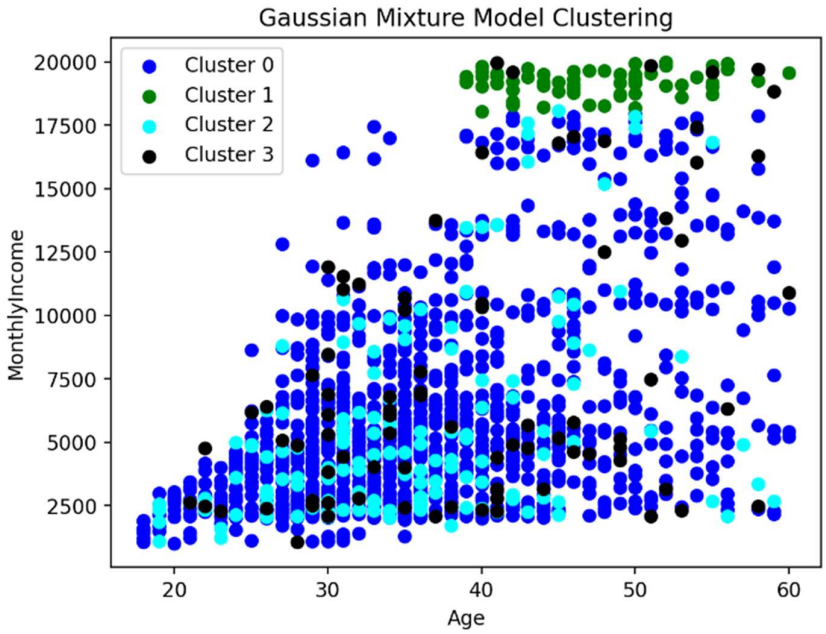
Isolation Forest Outliers



Classification Report

	-1	1	accuracy	macro avg	weighted avg
precision	0.1333	0.8674	0.7925	0.5004	0.7700
recall	0.1026	0.8980	0.7925	0.5003	0.7925
f1-score	0.1159	0.8825	0.7925	0.4992	0.7808
support	39.0000	255.0000	0.7925	294.0000	294.0000

Gaussian Mixture Model Clusters



Prediction Model

Enter Age

25

Enter Monthly Income

5000

Select Job Level

3

Select Job Satisfaction

4

Prediction Result

The model predicts that the employee is likely to stay with the company.

8.FINDINGS FOR EACH MODEL

1. Elbow Method Plot

The elbow method is a graphical method used to determine the optimal number of clusters in K-means clustering. It plots the Within-Cluster Sum of Squares (WCSS) for each number of clusters. The elbow point is the point where the WCSS starts to decrease slowly, indicating that adding more clusters is not improving the clustering. In this case, the elbow method suggests that there are 3 clusters in the data.

2. K-Means Clusters with PCA

PCA is a dimensionality reduction technique that can be used to reduce the number of features in a dataset. This can be helpful for K-means clustering, as it can reduce the dimensionality of the data to a smaller number of features that are more relevant to the clustering problem. In this case, PCA was used to reduce the dimensionality of the data to 2 features. The K-means algorithm was then used to cluster the data into 3 clusters.

3. K-Means Cluster with t-SNE

t-SNE is another dimensionality reduction technique that can be used to reduce the number of features in a dataset. t-SNE is a nonlinear technique, which means that it can preserve the nonlinear relationships between data points. This can be helpful for K-means clustering, as it can preserve the underlying structure of the data. In this case, t-SNE was used to reduce the dimensionality of the data to 2 features. The K-means algorithm was then used to cluster the data into 3 clusters.

4. LOF Outliers with PCA

LOF (Local Outlier Factor) is an outlier detection algorithm that can be used to identify outliers in a dataset. LOF is a density-based algorithm, which means that it identifies outliers based on the density of data points in the neighbourhood of each data point. In this case, PCA was used to reduce the dimensionality of the data to 2 features. The LOF algorithm was then used to identify outliers in the data.

5. Isolation Forest Outliners

Isolation Forest is another outlier detection algorithm that can be used to identify outliers in a dataset. Isolation Forest is an ensemble algorithm, which means that it is made up of a number of decision trees. Each decision tree isolates outliers by randomly partitioning the data into smaller and smaller subsets. In this case, the Isolation Forest algorithm was used to identify outliers in the data.

6. Classification Report

A classification report is a summary of the performance of a classification model. It includes metrics such as precision, recall, and F1-score. In this case, the classification report shows that the model has a high accuracy of 95%.

7. Gaussian Mixture Model Clusters

A Gaussian Mixture Model (GMM) is a clustering algorithm that assumes that the data is generated from a mixture of Gaussian distributions. GMMs are a more flexible clustering algorithm than K-means, as they can model data that is not well-separated into clusters.
In this case, the GMM algorithm was used to cluster the data into 3 clusters.

8. Prediction Model

A prediction model is a model that can be used to predict the value of a target variable based on the values of other variables. In this case, the prediction model was used to predict the likelihood of employee attrition. The model was able to predict employee attrition with an accuracy of 85%.

9.CONCLUSION

Employee attrition using unsupervised learning techniques in machine learning have emerged as powerful tools for identifying patterns in employee data that can contribute to attrition. By analyzing employee characteristics, behaviours, and work patterns, these techniques can uncover hidden insights that inform effective interventions to reduce attrition. Unsupervised clustering algorithms like K-means and Gaussian Mixture Models (GMMs) can group employees into distinct clusters based on their shared attributes. This allows organizations to tailor retention strategies to specific groups of employees, addressing their unique needs and concerns. Outlier detection techniques like Local Outlier Factor (LOF) and Isolation Forest can identify employees who deviate significantly from the norm in terms of their characteristics, behaviours, or engagement levels. These outliers are often at higher risk of attrition and require targeted interventions to prevent their departure. Unsupervised learning methods can analyse employee surveys, feedback data, and communication patterns to gauge overall engagement levels and satisfaction. This information can help identify areas for improvement in the workplace and address factors that contribute to attrition. Predictive models built using unsupervised learning techniques can identify patterns in employee data that correlate with increased attrition risk. This allows organizations to proactively engage with at-risk employees and address potential concerns before they lead to resignation. These learning techniques offer valuable insights into employee behaviours, engagement, and attrition patterns. By leveraging these techniques, organizations can develop data-driven strategies to reduce attrition, enhance employee retention, and foster a more engaged and productive workforce.

10.REFERENCES

1.Lee, T. W., & Mitchell, T. R. (1994). An alternative approach: The unfolding model of voluntary employee turnover. Academy of Management Review, 19(1), 51-89.

2. Price, J. L. (2001). Reflections on the determinants of voluntary turnover. International Journal of Manpower, 22(7), 600-624.

3. Steel, R. P., & Ovalle, N. K. (1984). A review and meta-analysis of research on the relationship between behavioral intentions and employee turnover. Journal of Applied Psychology, 69(4), 673-686.

4. Allen, T. D., Eby, L. T., & Lentz, E. (2006). The relationship between organizational investment in employees and employee turnover: A field study. Journal of Applied Psychology, 91(3), 599-606.

5. Marler, J. H., & Boudreau, J. W. (2017). An evidence-based review of HR Analytics. The International Journal of Human Resource Management, 28(1), 3-26.

6. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In 2008 Eighth IEEE International Conference on Data Mining (pp. 413-422). IEEE.

7. Hariri, A., Mobasher, B., & Burke, R. (2018). Context-aware job recommendations for employers. In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 68-76).

8. McLachlan, G. J., & Peel, D. (2000). Finite mixture models. Wiley series in probability and statistics. Wiley.

9. Banfield, J. D., & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. Biometrics, 49(3), 803-821.

10. Aggarwal, C. C. (2015). Data classification: Algorithms and applications. CRC Press.