EMPLOYEE ATTRITION PREDICTION

Presented by :-
Vani Tyagi (49)
Nasreen Parween (58)
Sawan Bagle (39)

AIM

The aim of employee attrition prediction is to identify which employees are likely to leave an organization so that proactive measures can be taken to retain them. Predicting attrition helps companies reduce costs associated with hiring and training new employees, maintain team stability, and ensure productivity by keeping experienced and skilled employees.

This study compares **K means Clustering , Naive Bayes Classificatio**n , and **Decision Tree** models for their predictive accuracy and performance.

OVERVIEW

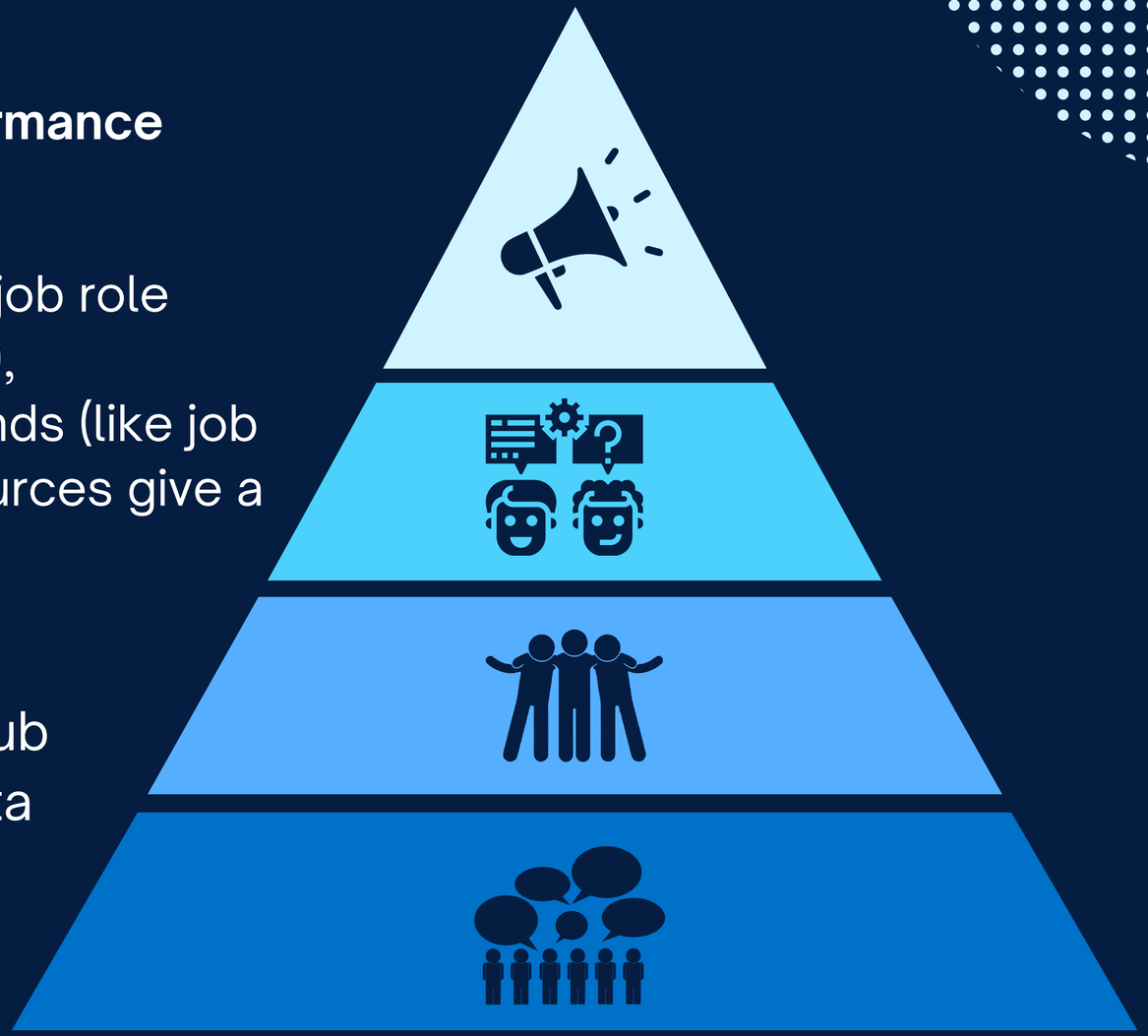
In today's competitive market, employee attrition poses significant challenges for organizations. Understanding the reasons behind attrition is crucial for developing effective retention strategies. This presentation explores methods to predict attrition and enhance employee engagement.

In this project, we will be dealing with binary class classification which are-

0- No attrition and 1 - attrition

ABOUT THE DATASET

- **IBM HR Analytics Employee Attrition & Performance Dataset is used.**
- Features are computed from HR records (like job role and tenure), surveys (like engagement scores), behavioral data (like attendance), external trends (like job market data), and manager insights. These sources give a well-rounded view to predict who might leave.
- Dataset Kaggle Link - https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/data



ATTRIBUTE INFORMATION -

Age, Attrition, BusinessTravel, DailyRate, Department, DistanceFromHome, Education, EducationField, EnvironmentSatisfaction, Gender, HourlyRate, JobInvolvement, JobLevel, JobRole, JobSatisfaction, MaritalStatus, MonthlyIncome, MonthlyRate, NumCompaniesWorked, 'OverTime, PercentSalaryHike, PerformanceRating, RelationshipSatisfaction, StockOptionLevel, TotalWorkingYears, TrainingTimesLastYear, WorkLifeBalance, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, 'YearsWithCurrManager

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	J
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	3	Male	41	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	4	Male	42	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	2	Male	87	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	4	Male	63	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	2	Male	82	

1470 rows × 35 columns

WORK FLOW

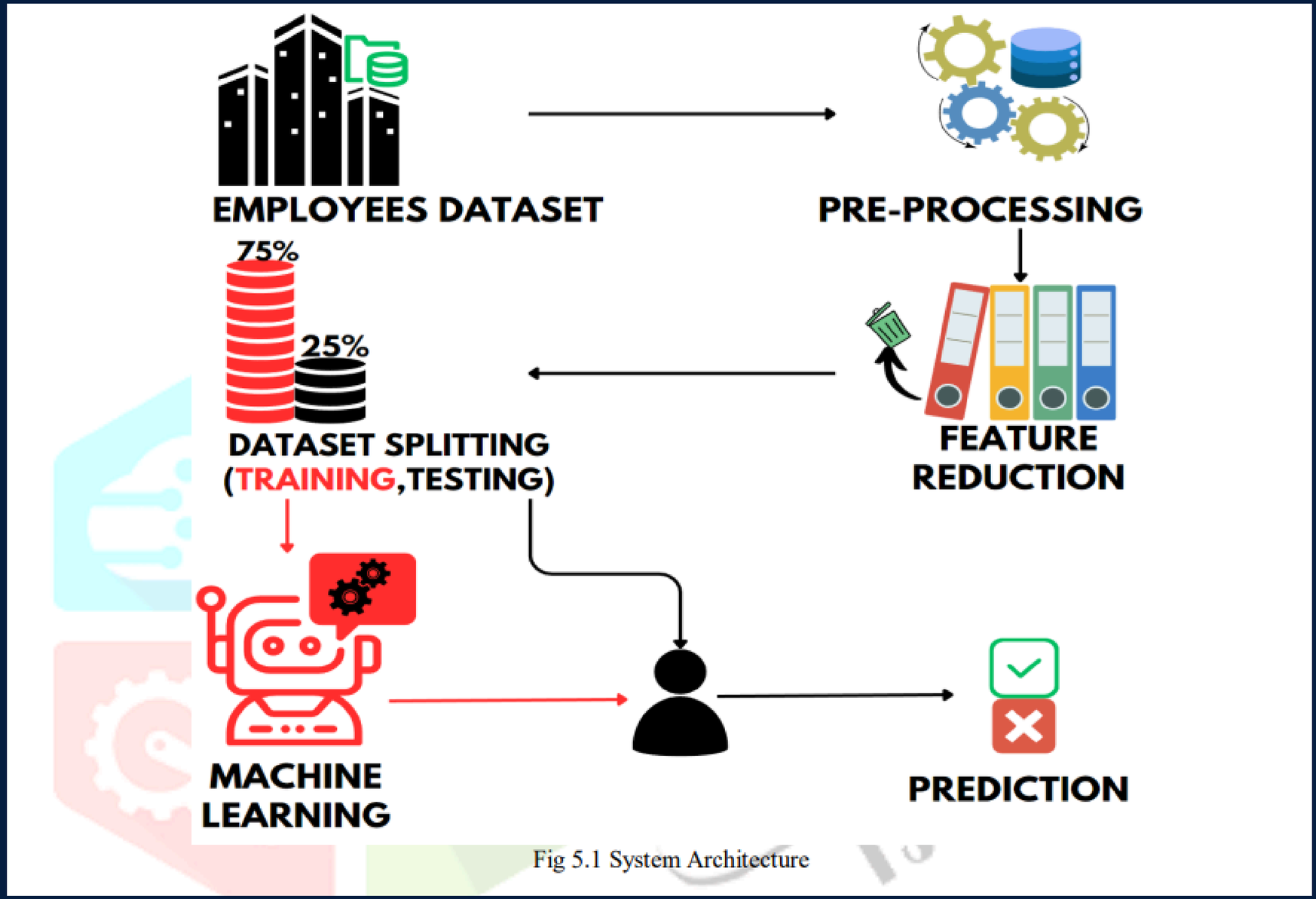


Fig 5.1 System Architecture

IMPLEMENTATION

Libraries Used:

- Numpy
- Matplotlib
- pandas
- seaborn
- sklearn
- hvplot

Preprocessing Steps:

Raw data is converted into an understandable form and made ready for further analysis. This step includes data cleaning, handling missing values, transformation, feature scaling etc.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
```

HANDLING NULL VALUES

Luckily we have non-null values in our dataset, that saves time for diving deep in analysis of our data.
We have total of 35 columns in our dataset,in which we have 26 integer based columns, and rest are objects

LABEL ENCODING

Label encoding is a technique used to convert categorical variables into numerical format.

sklearn.preprocessing library can be used for the same.

For the sake of convenience, we will be encoding the Attrition label as-
-> 1 for Attrition
-> 0 for No Attrition

This can be done by using Label Encoder and Fit_transform function from the sklearn.preprocessing module.

	Age	Attrition
0	41	1
1	49	0
2	37	1
3	33	0
4	27	0

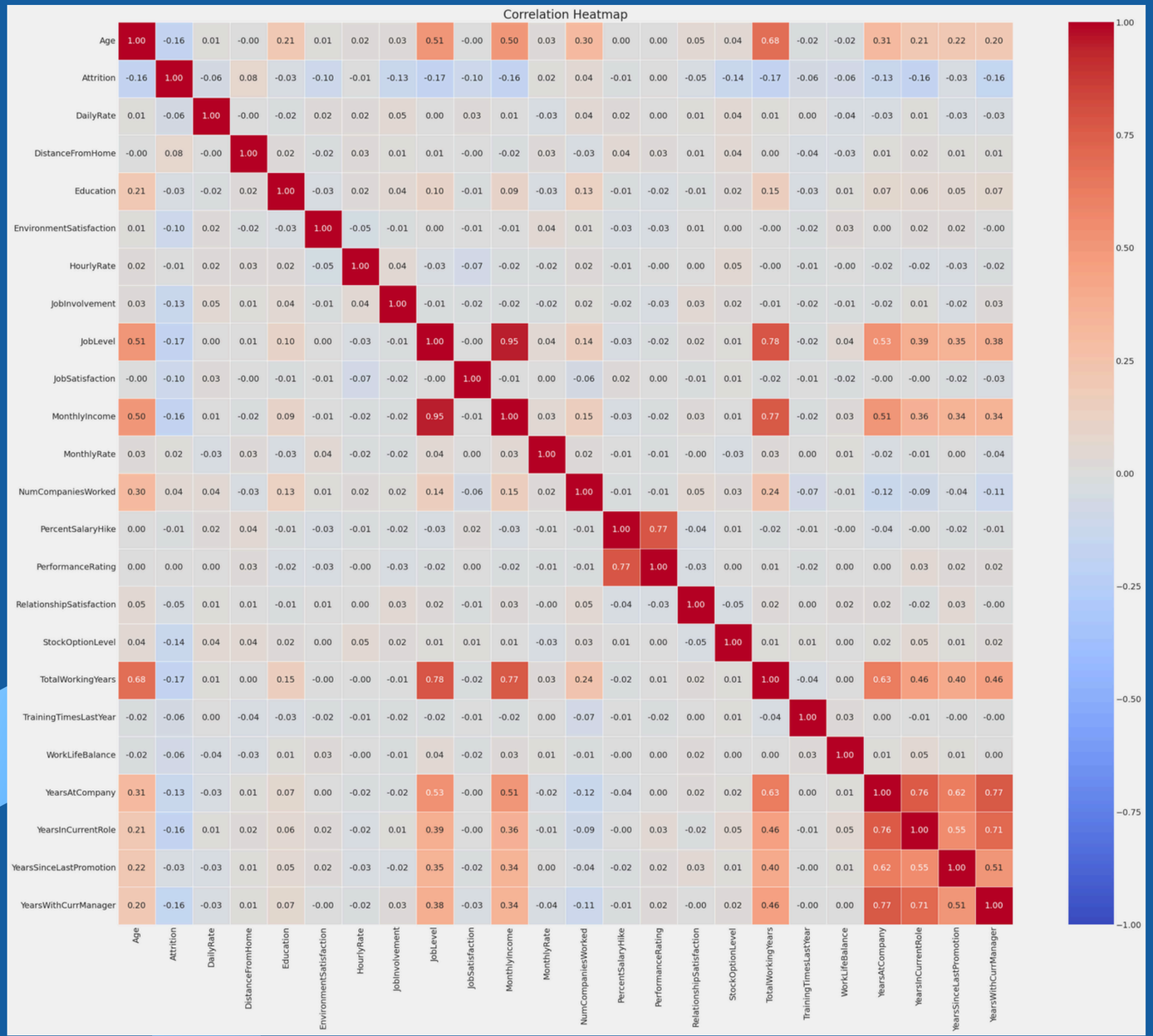
Finding the correlation between different pairs of attributes

Correlation refers to the statistical relationship between two or more variables where the variation in one variable is associated with variation in another variable.

	Age	Attrition	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfaction	MonthlyIncome	Month
Age	1.000000	-0.159205	0.010661	-0.001686	0.208034	0.010146	0.024287	0.029820	0.509604	-0.004892	0.497855	0.(
Attrition	-0.159205	1.000000	-0.056652	0.077924	-0.031373	-0.103369	-0.006846	-0.130016	-0.169105	-0.103481	-0.159840	0.(
DailyRate	0.010661	-0.056652	1.000000	-0.004985	-0.016806	0.018355	0.023381	0.046135	0.002966	0.030571	0.007707	-0.(
DistanceFromHome	-0.001686	0.077924	-0.004985	1.000000	0.021042	-0.016075	0.031131	0.008783	0.005303	-0.003669	-0.017014	0.(
Education	0.208034	-0.031373	-0.016806	0.021042	1.000000	-0.027128	0.016775	0.042438	0.101589	-0.011296	0.094961	-0.(
...	...	...	...	...	...	...	...	...	...	...	...	
WorkLifeBalance	-0.021490	-0.063939	-0.037848	-0.026556	0.009819	0.027627	-0.004607	-0.014617	0.037818	-0.019459	0.030683	0.(
YearsAtCompany	0.311309	-0.134392	-0.034055	0.009508	0.069114	0.001458	-0.019582	-0.021355	0.534739	-0.003803	0.514285	-0.(
YearsInCurrentRole	0.212901	-0.160545	0.009932	0.018845	0.060236	0.018007	-0.024106	0.008717	0.389447	-0.002305	0.363818	-0.(
YearsSinceLastPromotion	0.216513	-0.033019	-0.033229	0.010029	0.054254	0.016194	-0.026716	-0.024184	0.353885	-0.018214	0.344978	0.(
YearsWithCurrManager	0.202089	-0.156199	-0.026363	0.014406	0.069065	-0.004999	-0.020123	0.025976	0.375281	-0.027656	0.344079	-0.(

24 rows × 24 columns

HEATMAP FOR CORRELATION-

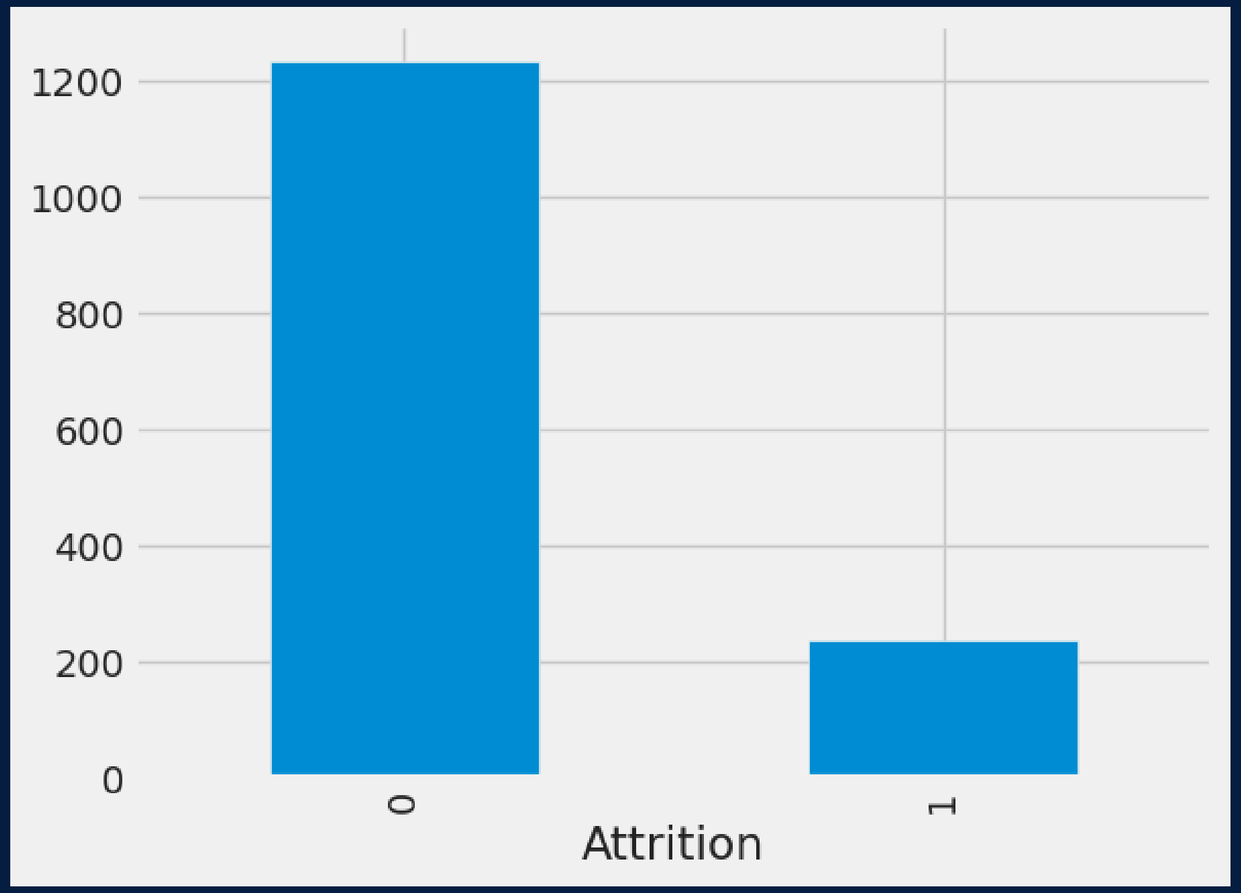


Heatmap is a visual representation of data where individual values contained in a matrix are represented as colors.

Heatmaps are commonly used to show the concentration or intensity of data in a two-dimensional format, making patterns, correlations, or outliers in data more visually accessible.

SOME VISUAL REPRESENTATIONS

This plot shows the number of instances for no attrition (0) and attrition (1).



Before Sampling

Attrition	count
0	1233
1	237

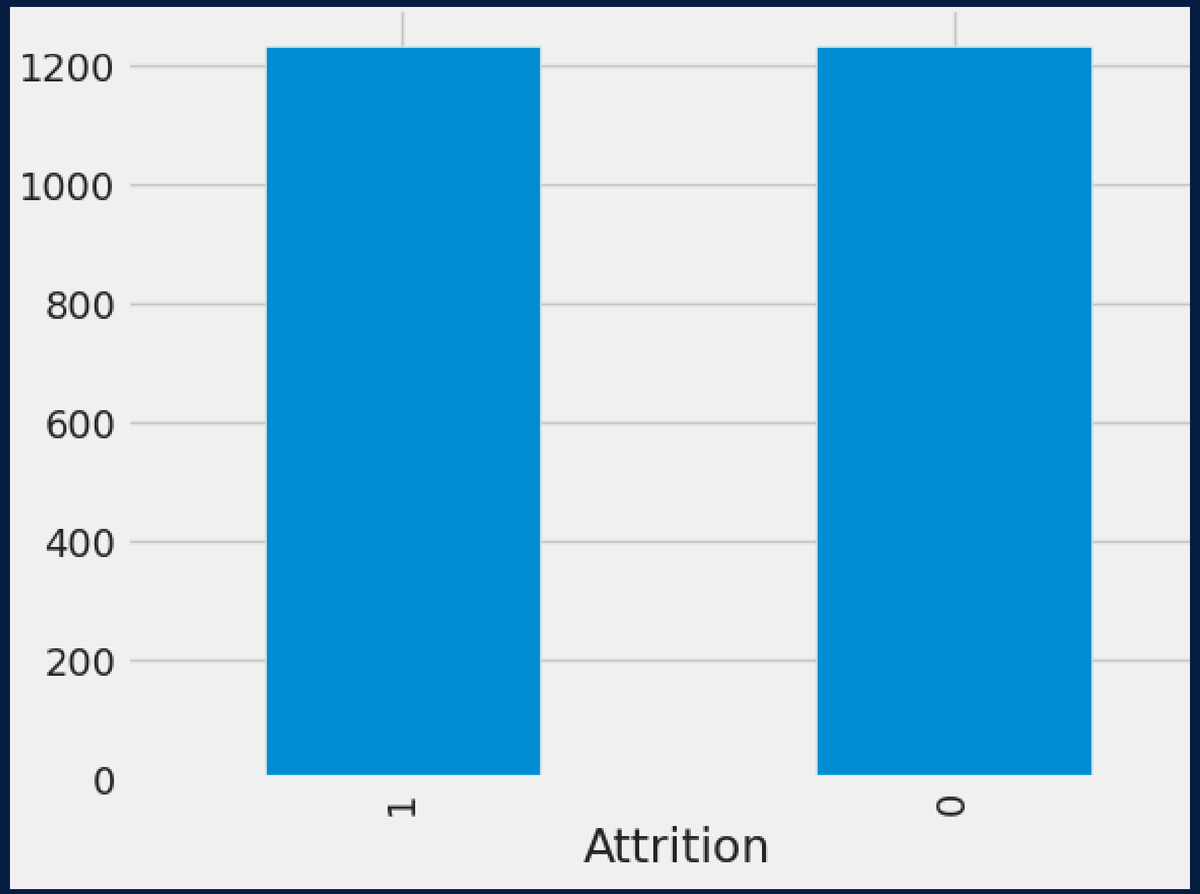
dtype: int64

No Attrition (0) = 1233
Attrition (1) = 237

This show that our datasets are very imbalanced . So we can use SMOTE sampling to resolve the problem of imbalanced datasets.

OVERSAMPLING

SMOTE (**Synthetic Minority Over-sampling Technique**)
It is a widely used technique in data mining to address the issue of imbalanced datasets.



After Sampling

Purpose of SMOTE

- **Balancing Classes:** SMOTE helps to create a more balanced dataset by increasing the representation of the minority class, improving model performance on this class.
- **Avoiding Overfitting:** Traditional oversampling can lead to overfitting. SMOTE reduces this risk by generating synthetic examples rather than simple copies.

Attrition	count
1	1233
0	1233

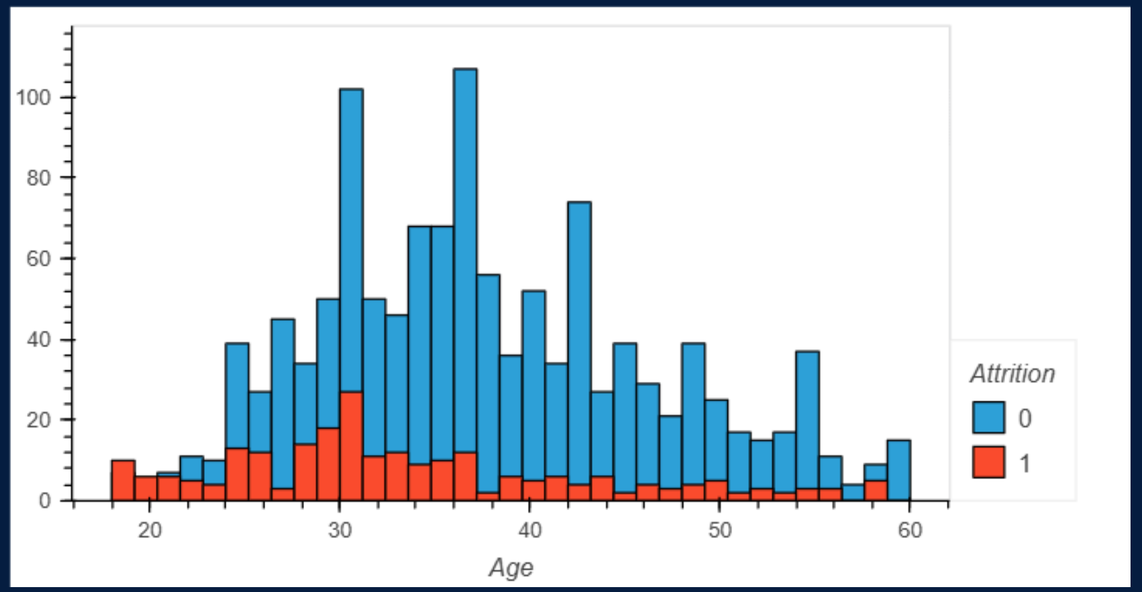
dtype: int64

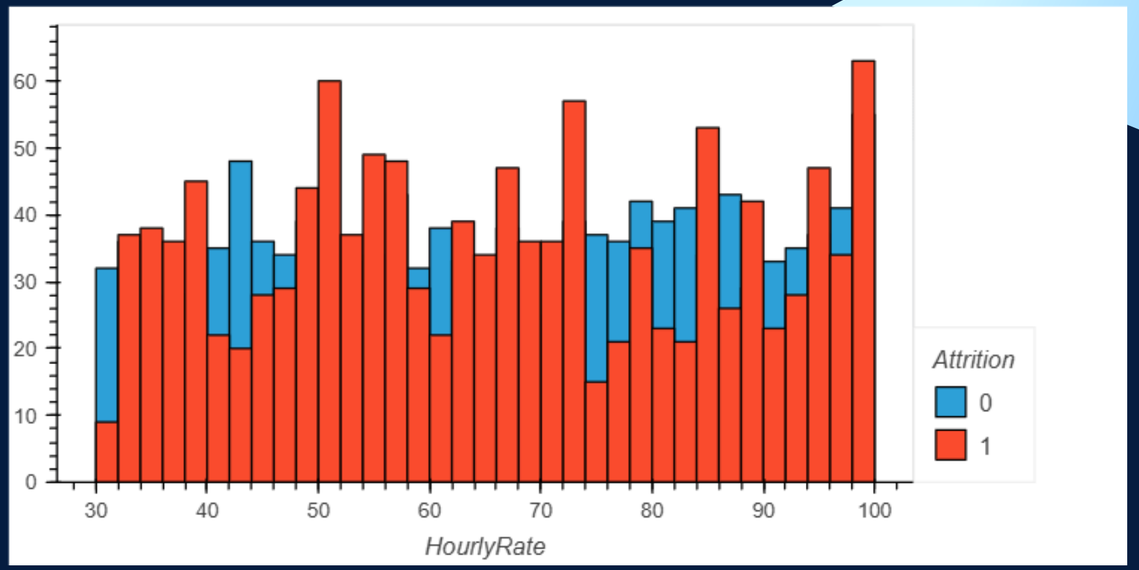
No Attrition (0) = 1233
Attrition (1) = 1233

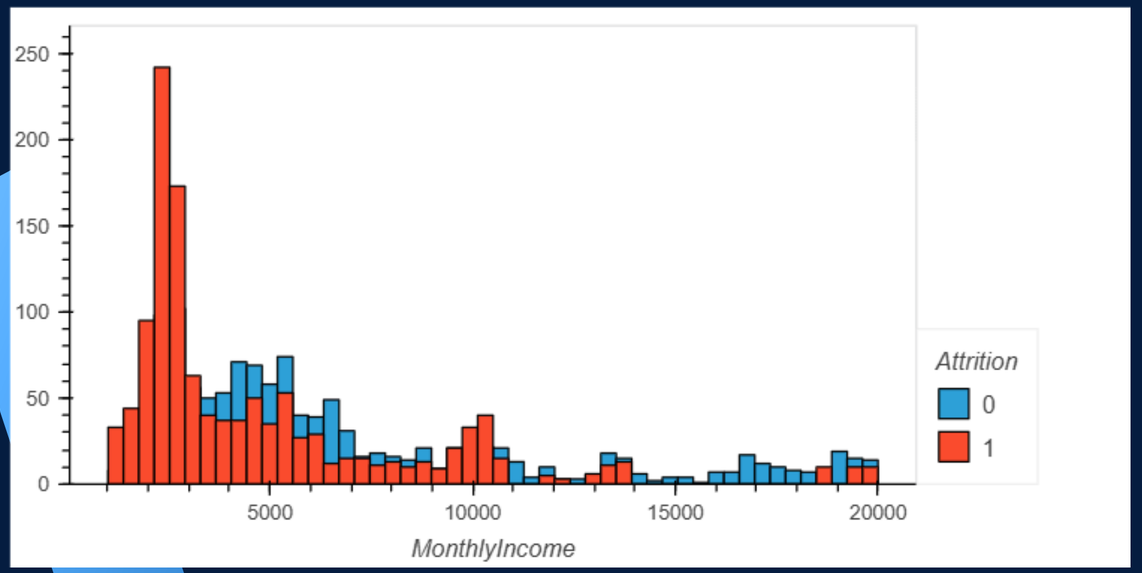
EXPLORATORY DATA ANALYSIS

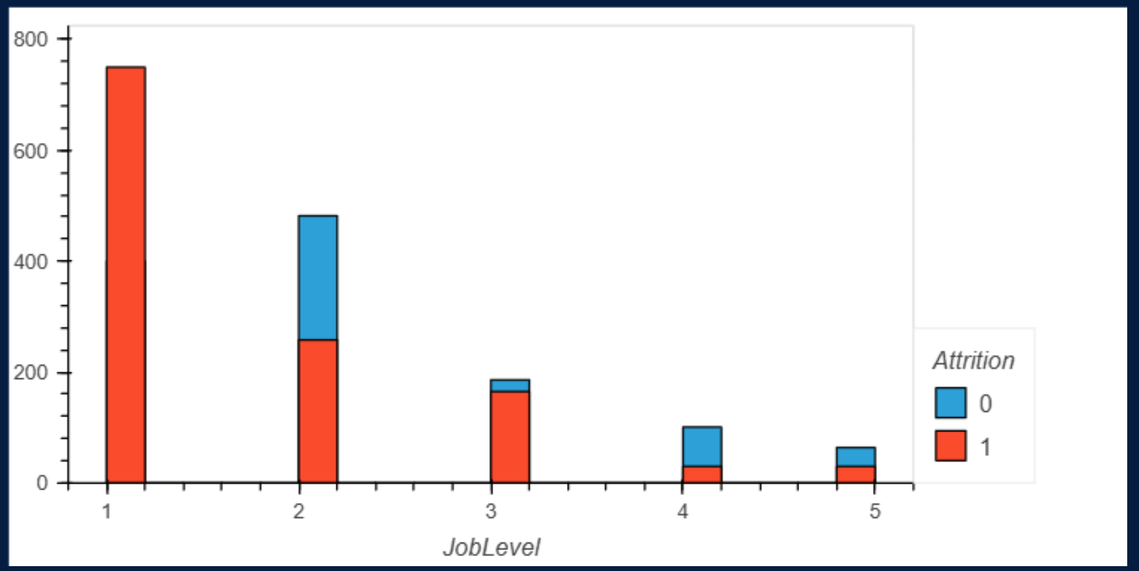
- EDA is a critical initial step in the data analysis process where analysts use statistical tools and visualizations to understand the main characteristics and patterns within a dataset.
- EDA is a foundational phase in data science that involves a holistic exploration of the dataset.
- EDA helps in uncovering trends, spotting anomalies, testing assumptions, and identifying relationships between variables
- It provides a snapshot of data’s central tendencies

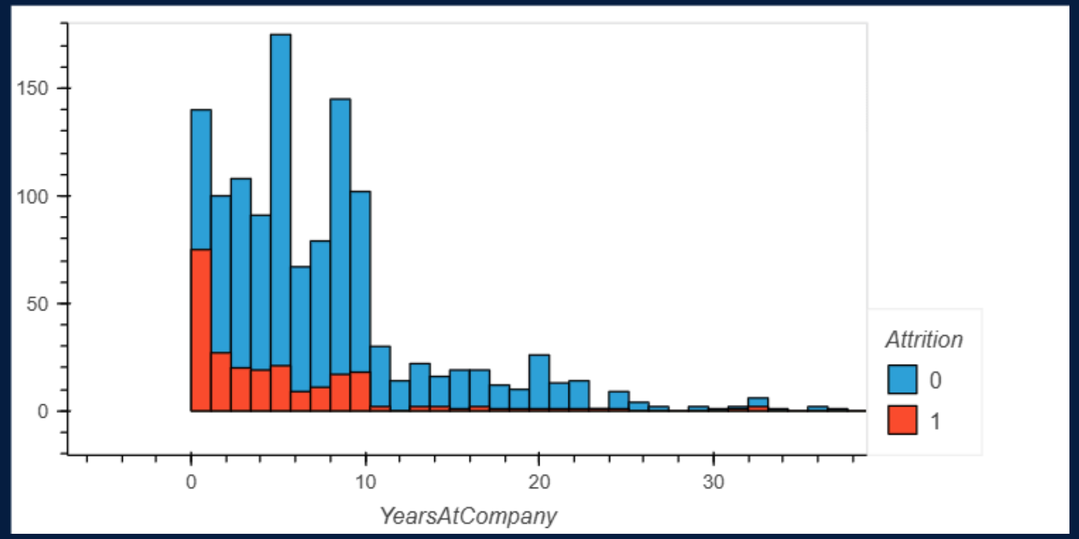
VISUALIZATION

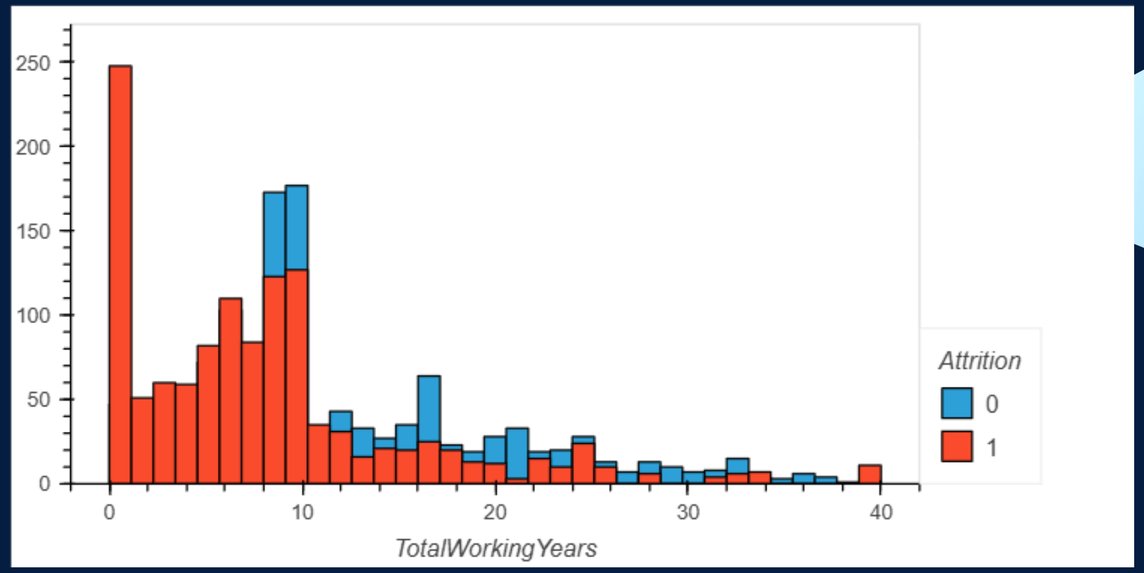


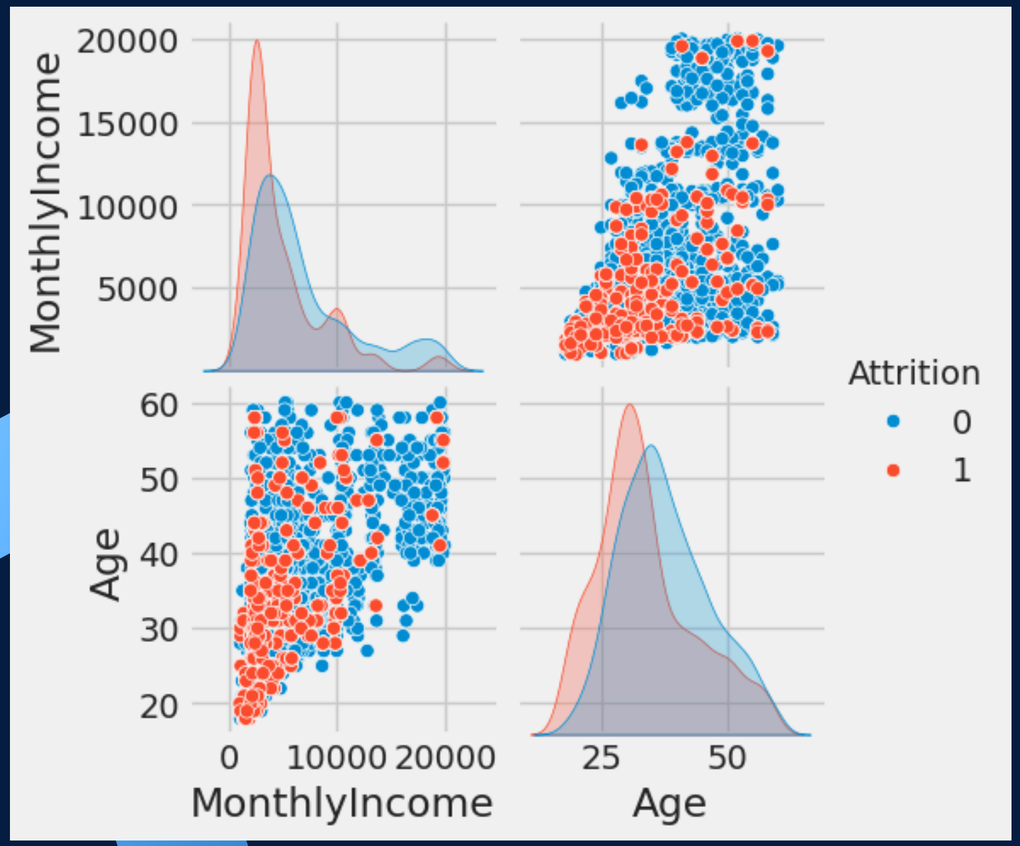


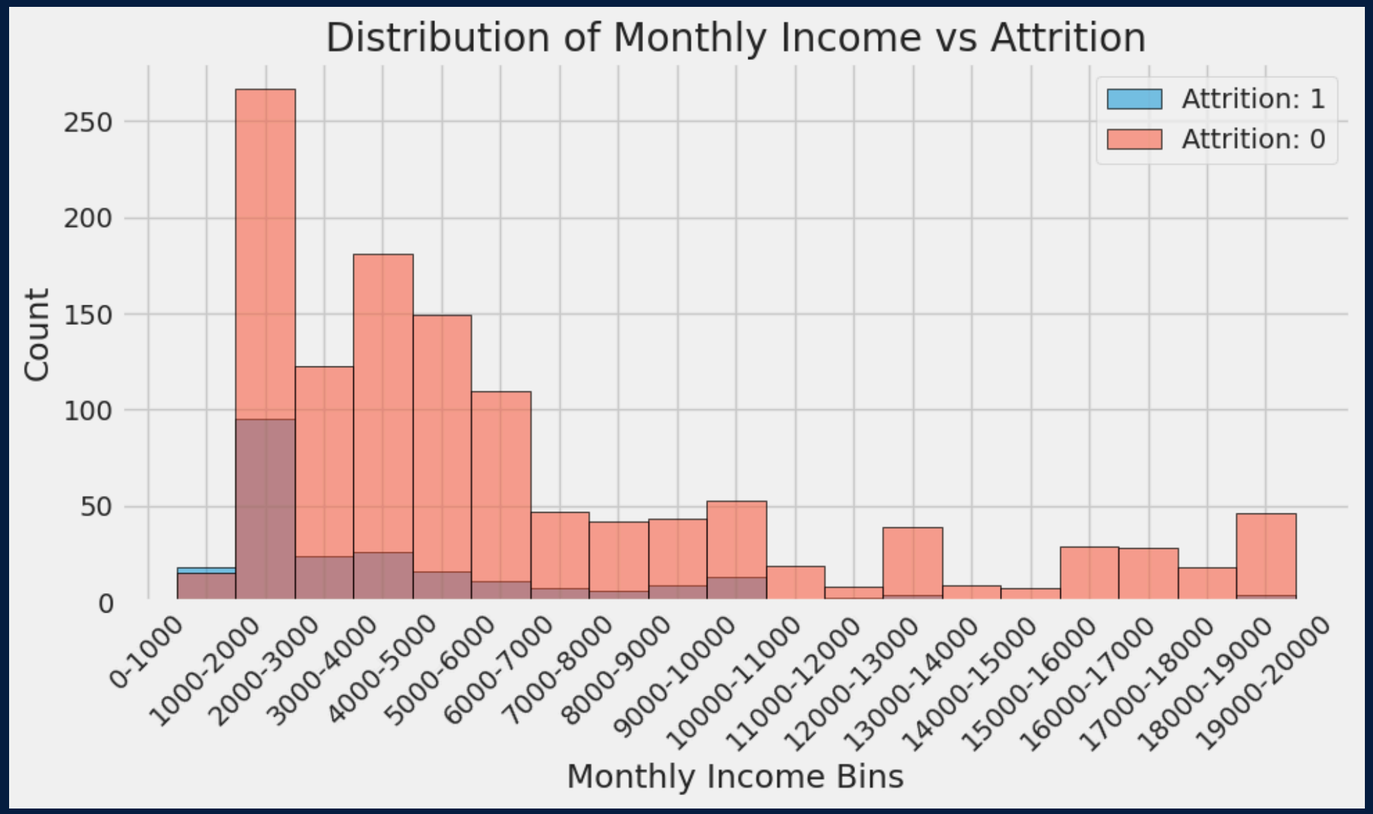





150
100
50
0
0
10
20
30
YearsAtCompany
Attrition
0
1


250
200
150
100
50
0
0
10
20
30
40
TotalWorkingYears
Attrition
0
1


MonthlyIncome
20000
15000
10000
5000
Age
60
50
40
30
20
0
10000
20000
MonthlyIncome
25
50
Age
Attrition
0
1


Distribution of Monthly Income vs Attrition
Attrition: 1
Attrition: 0
Count
250
200
150
100
50
0
0-1000
1000-2000
2000-3000
3000-4000
4000-5000
5000-6000
6000-7000
7000-8000
8000-9000
9000-10000
10000-11000
11000-12000
12000-13000
13000-14000
14000-15000
15000-16000
16000-17000
17000-18000
18000-19000
19000-20000
Monthly Income Bins

Our Takeaways from visualization

- The workers with low JobLevel, MonthlyIncome, YearAtCompany, and TotalWorkingYears are more likely to quit there jobs.
- **BusinessTravel :** The workers who travel alot are more likely to quit then other employees.
- **Department :** The worker in Research & Development are more likely to stay then the workers on other departement.
- **EducationField :** The workers with Human Resources and Technical Degree are more likely to quit then employees from other fields of educations.
- **Gender :** The Male are more likely to quit.
- **JobRole :** The workers in Laboratory Technician, Sales Representative, and Human Resources are more likely to quit the workers in other positions.
- **MaritalStatus :** The workers who have Single marital status are more likely to quit the Married, and Divorced.
- **OverTime :** The workers who work more hours are likely to quit then others.

FEATURE ENCODING

Feature encoding is a way to convert non-numeric data, like words or categories, into numbers so that a machine learning model can understand it.

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole
0	41	2	1102	2	1	2	1	2	0	94	3	2	7
1	49	1	279	1	8	1	1	3	1	61	2	2	6
2	37	2	1373	1	2	2	4	4	1	92	2	1	2
3	33	1	1392	1	3	4	1	4	0	56	3	1	6
4	27	2	591	1	2	1	3	1	1	40	3	1	2

Splitting the data and feature scaling

Split the data set into X and Y, where X is independent variables and Y is Dependent variable.

```
# Import train_test_split
from sklearn.model_selection import train_test_split
# Create train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

- Feature Scaling- process of normalizing or standardization of the range of features in the dataset.
- Standardization is a scaling technique wherein we convert data into the below format:Mean = 0 and standard deviation = 1
- Z=(x-xmean)/standard deviation

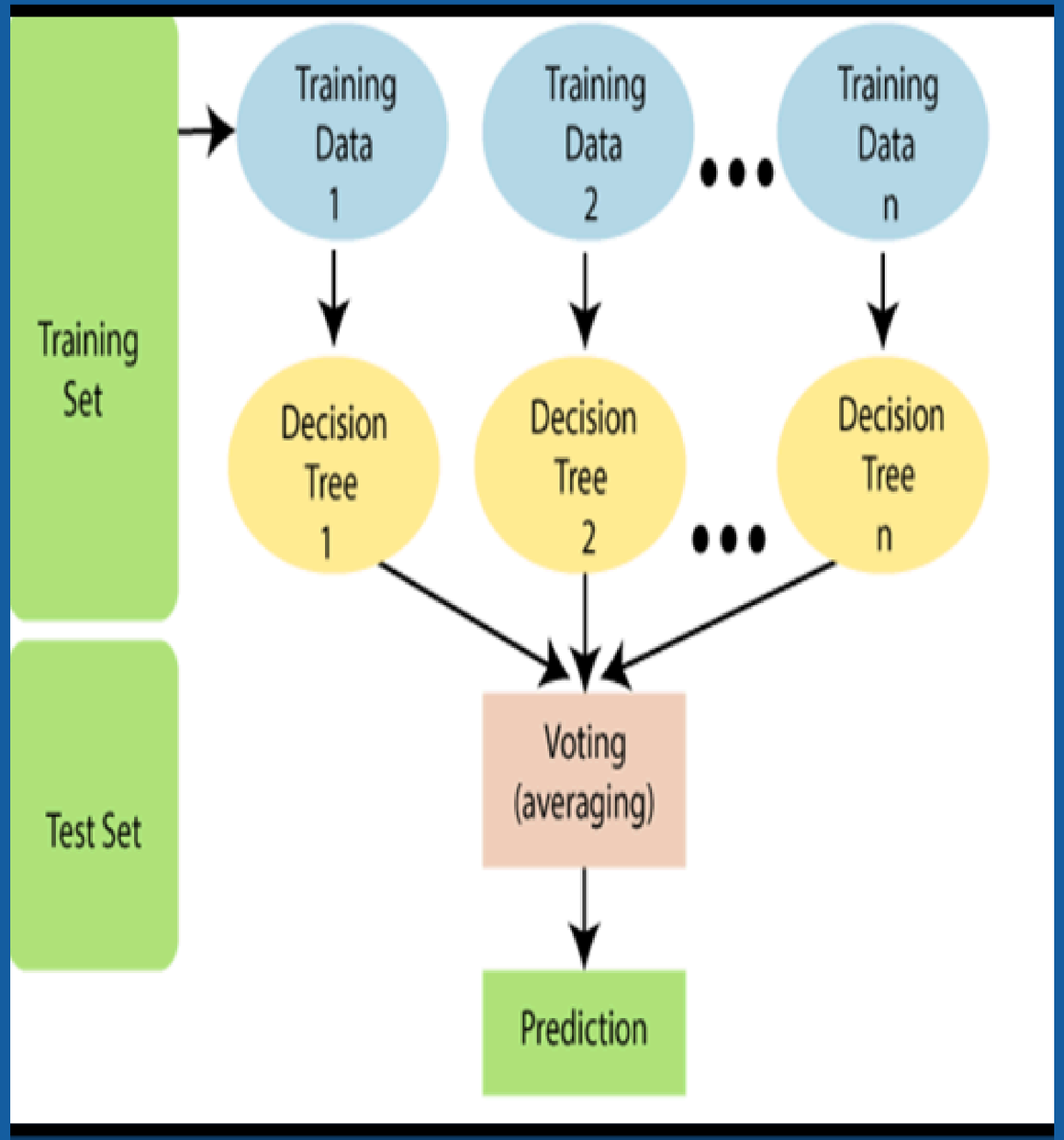
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

MODEL TRAINING

We have applied 3 classification model:

1. **Random Forest**
2. **Naive Bayes Classification**
3. **Decision Tree**

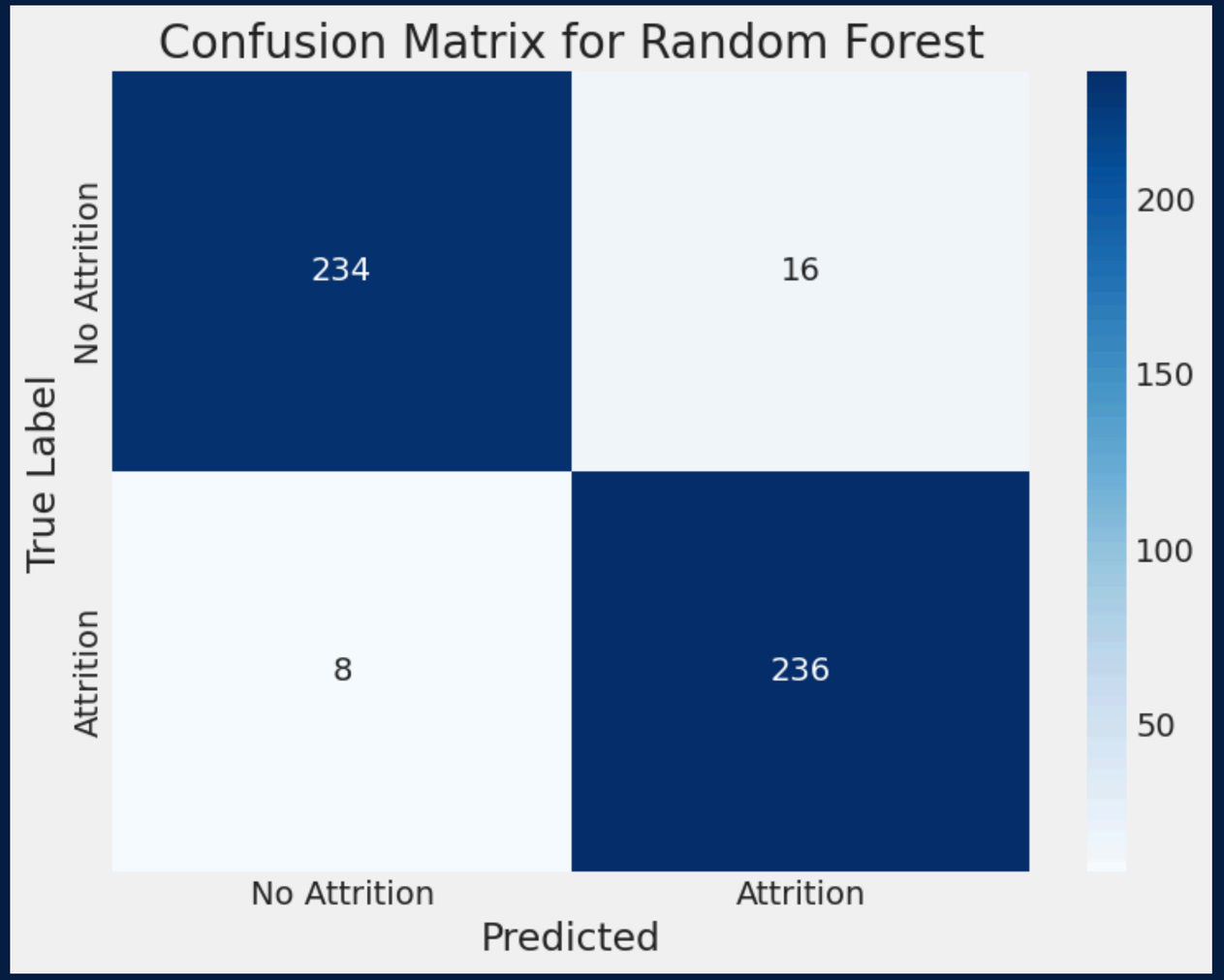
1. RANDOM FOREST



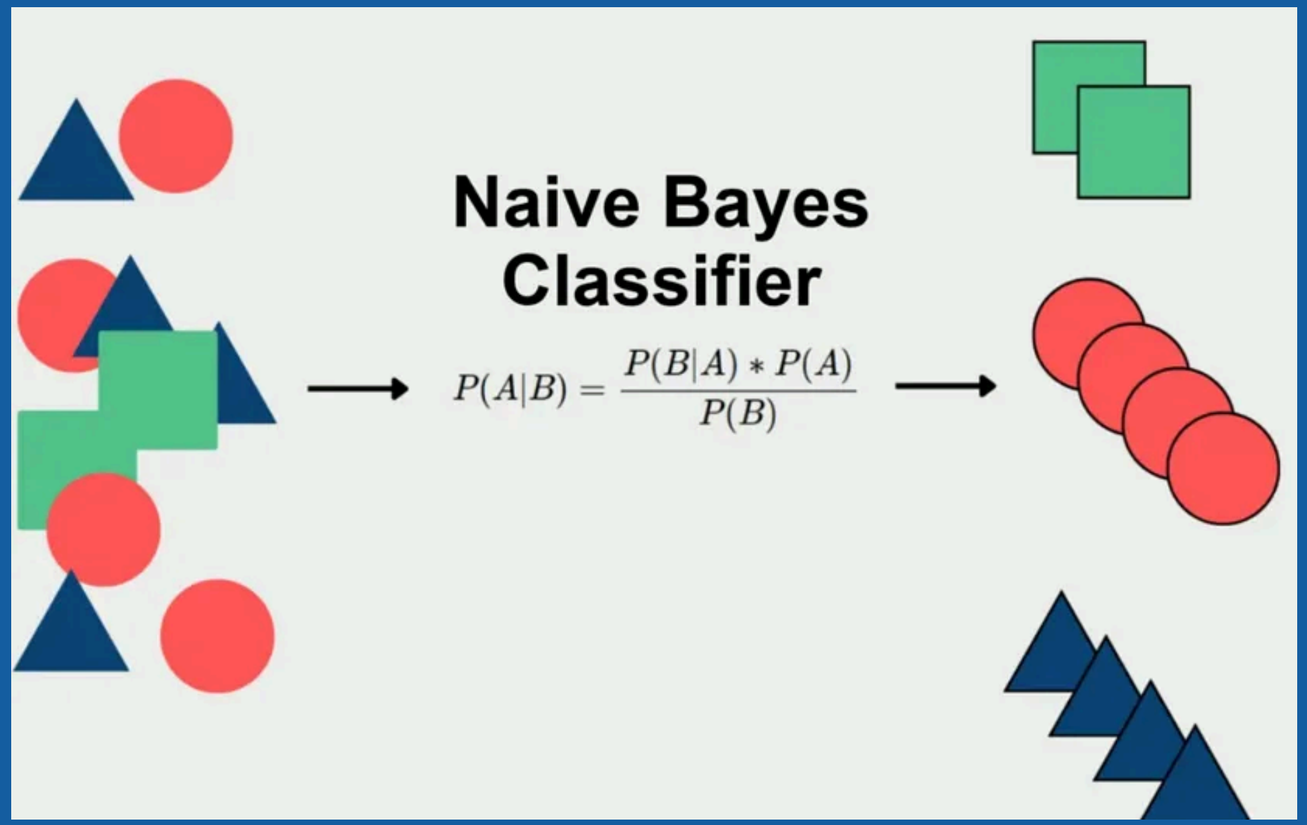
- Random Forest is an ensemble learning method that constructs a multitude of decision trees and merges them to improve predictive accuracy.
- These trees then make individual predictions, and the final prediction is determined by aggregating the predictions of all trees, often using majority voting for classification or averaging for regression.
- This helps reduce overfitting and improve generalization.
- It provides feature importance and can handle missing values.
- It is robust to noise and outliers in the data.

RESULTS ANALYSIS OF MODEL -1

```
Accuracy of Random Forest: 95.1417004048583%
```



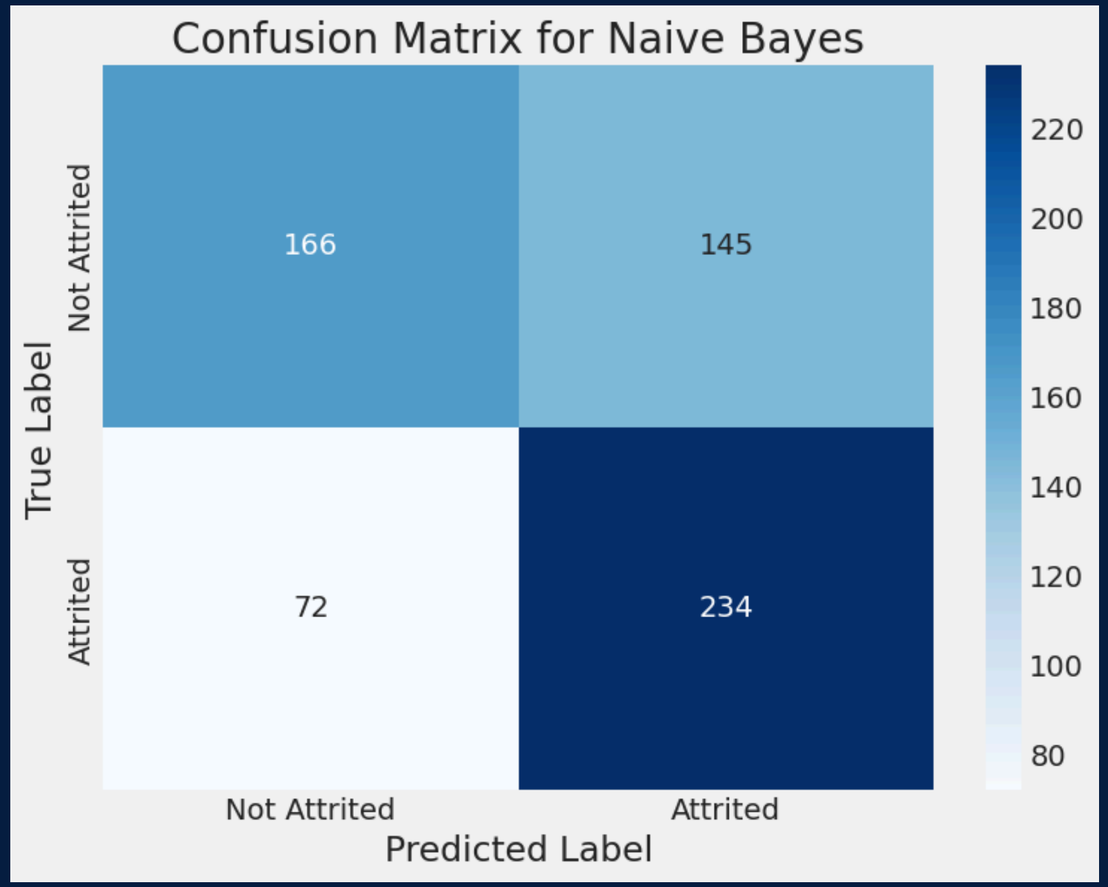
2. NAIVE BAYES CLASSIFICATION



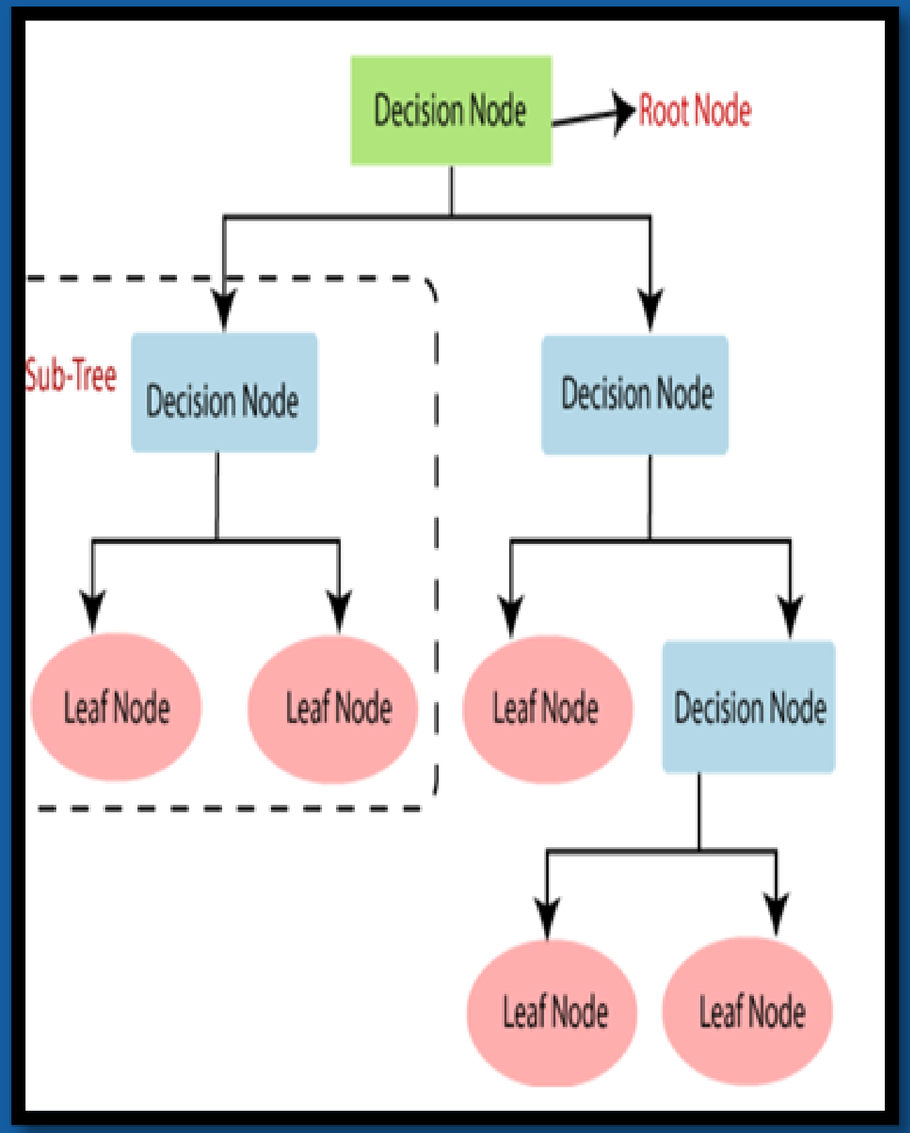
- Naive Bayes Classification is a method that predicts the category of something (like whether an email is spam or not) based on probability.
- It uses the idea that each feature (like words in an email) independently affects the outcome.
- It calculates the probability for each possible category and picks the one with the highest chance.
- Naive Bayes is fast and works well for tasks like spam filtering or sorting text.

RESULTS ANALYSIS OF MODEL -2

```
Accuracy of Naive Bayes: 64.82982171799027 %
```



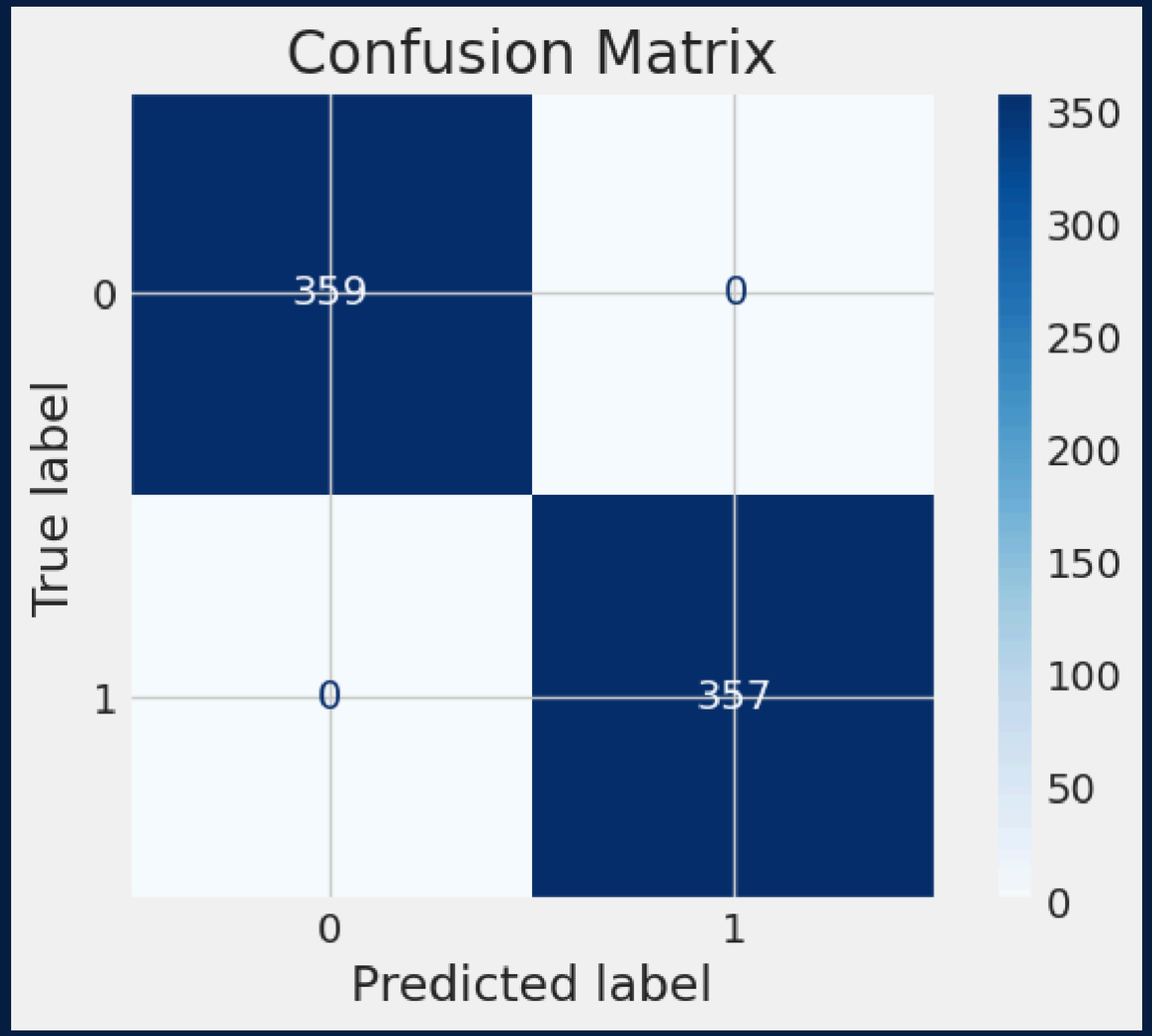
3. DECISION TREE MODEL



- Decision Tree is a tree-like model that uses a branching method to illustrate every possible outcome of a decision. It is easy to interpret and suitable for complex datasets.
- Nodes in a decision tree represent features and branches represent decisions.
- The algorithm splits the dataset based on the best attribute at each node.
- They Capture both linear and non – linear relationships in the data.

RESULTS ANALYSIS OF MODEL -3

Accuracy of Decision tree: 100.0 %

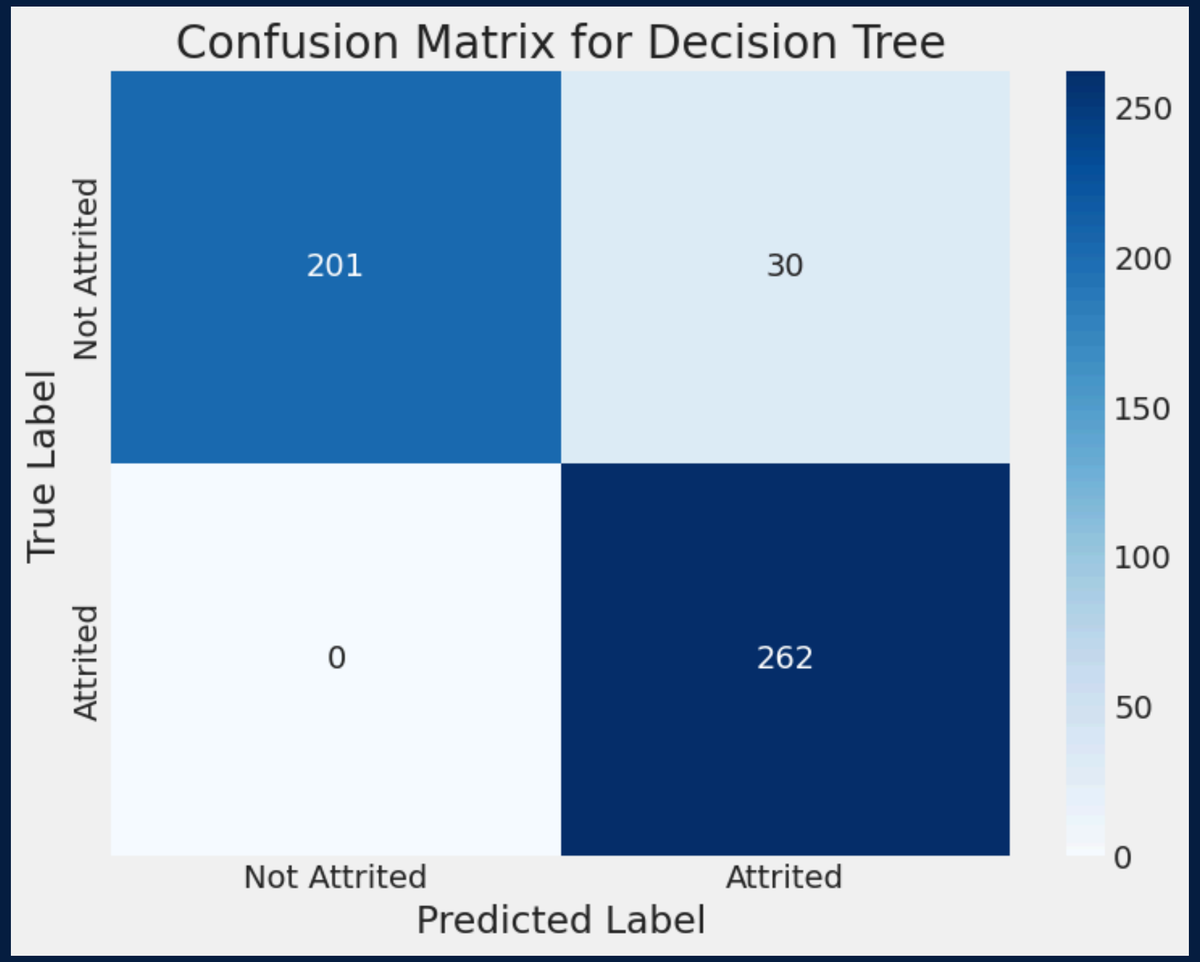


	precision	recall	f1-score	support
0	1.00	1.00	1.00	359
1	1.00	1.00	1.00	357
accuracy			1.00	716
macro avg	1.00	1.00	1.00	716
weighted avg	1.00	1.00	1.00	716

Since, we are getting high accuracy for training data, we can say that our model is over fit.
For that we will be using unseen data and k fold cross validation.

AFTER APPLYING K-FOLD CROSS VALIDATION

```
Decision Tree
Average accuracy across 5 folds: 92.70162846654786%
```



For Decision Tree Classifier-
K=5

CLASSIFICATION REPORT OF THE MODELS ON TEST DATA -

	precision	recall	f1-score	support
0	0.97	0.94	0.95	250
1	0.94	0.97	0.95	244
accuracy			0.95	494
macro avg	0.95	0.95	0.95	494
weighted avg	0.95	0.95	0.95	494

Model 1- Random Forest

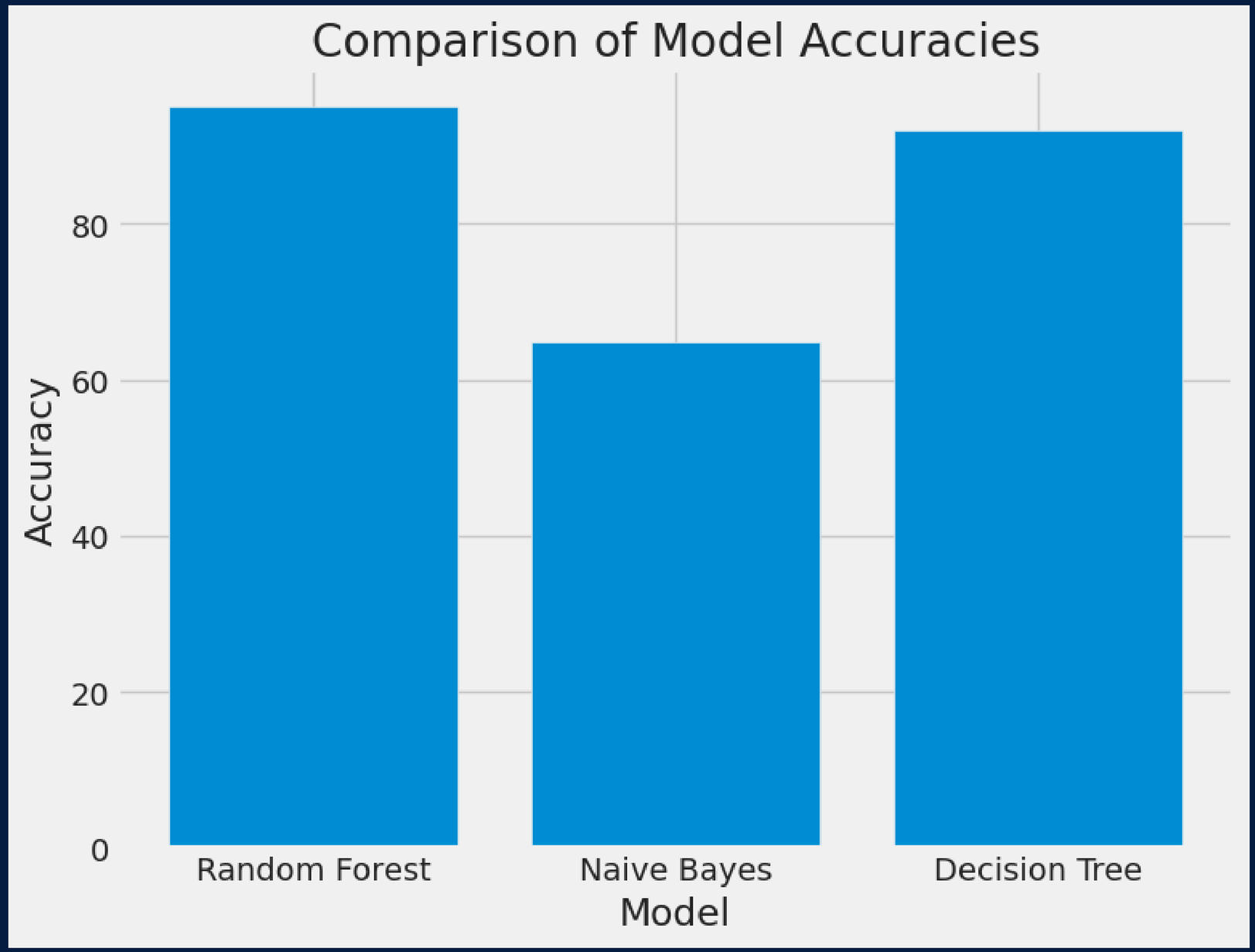
	precision	recall	f1-score	support
0	0.70	0.53	0.60	311
1	0.62	0.76	0.68	306
accuracy			0.65	617
macro avg	0.66	0.65	0.64	617
weighted avg	0.66	0.65	0.64	617

Model 2- Naive Bayes Classification

	precision	recall	f1-score	support
0	1.00	0.87	0.93	231
1	0.90	1.00	0.95	262
accuracy			0.94	493
macro avg	0.95	0.94	0.94	493
weighted avg	0.95	0.94	0.94	493

Model 3- Decision Tree

COMPARISON OF MODEL ACCURACIES



CONCLUSION

For predicting employee attrition, **Random Forest, Decision Tree,** and **Naive Bayes** models vary in effectiveness depending on the nature of the dataset:

Random Forest:
Gave the best overall results with high accuracy and balanced performance. It's a strong model for prediction when the data is complex.

Decision Tree:
Also performed very well and is easy to understand. It makes clear decisions and is good for real-world use, though slightly behind Random Forest in balance.

Naive Bayes:
Is simple and fast, but had lower accuracy. It works better when features are independent, which may not always be true for employee data.

Random Forest is the most accurate, Decision Tree is easy to interpret and effective, and Naive Bayes is quick but less reliable for this task.

Future Research Directions

- Exploring psychological factors and economic conditions in future research for even more robust predictions.
- Adding factors like job satisfaction and economic conditions for even more precise predictions in future models.
- Enable real-time monitoring for timely interventions and Develop models adaptable across industries.
- Use advanced algorithms like deep learning for accuracy

References

- https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/data
- Najafi-Zangeneh, S.; Shams-Gharneh, N.; Arjomandi-Nezhad, A.; Hashemkhani Zolfani, S. An Improved Machine Learning-Based Employees Attrition Prediction Framework with Emphasis on Feature Selection. Mathematics 2021, 9, 1226. https://doi.org/10.3390/ math9111226
- Raza, A.; Munir, K.; Almutairi, M.; Younas, F.; Fareed, M.M.S. Predicting Employee Attrition Using Machine Learning Approaches. Appl. Sci. 2022, 12, 6424. https://doi.org/10.3390/ app12136424
- Francesca Fallucchi , Marco Coladangelo and Ernesto William De Luca, Predicting Employee Attrition Using Machine Learning Techniques https://doi.org/10.3390/computers9040086

Thanks!