Employee Churn Prediction

A Project Report

Submitted in partial fulfillment of the requirements

Of

……. Track Name ……

by

SUBASHAN.K , subash31052004@gmail.com

Under the Guidance of

P.Raja, Master Trainer

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all the teachers who have supported and guided me throughout the course of this project. First and foremost, I would like to thank [Teacher's Name], whose expertise, encouragement, and constant support helped shape the direction of this work. Their insightful feedback and constructive criticism have been invaluable.

I am also deeply grateful to [Teacher's Name] for their unwavering patience and guidance, which allowed me to explore my ideas in depth and to overcome challenges with confidence. A special thanks to [Teacher's Name] for their inspiration and for making complex concepts more accessible.

Finally, I extend my appreciation to all the faculty members whose dedication to education and mentorship has contributed to my academic growth. Without their continued support, this project would not have been possible.

Thank you all for your commitment to nurturing my curiosity and helping me reach this milestone.

****

ABSTRACT

Employee churn prediction is a critical task for organizations aiming to retain talent and optimize workforce management. This project focuses on developing a machine learning model to predict employee churn, helping organizations proactively identify employees at risk of leaving. The objective is to build a data-driven solution that can assist HR departments in making informed decisions regarding employee retention strategies, resource allocation, and overall organizational efficiency.

The project utilizes a dataset containing various employee attributes, such as demographics, job role, tenure, compensation, performance, and work environment factors. Through exploratory data analysis (EDA), we identify key features that influence employee turnover and assess their relationships with employee retention. Several machine learning algorithms, including logistic regression, decision trees, random forests, and support vector machines, are applied to develop predictive models.

Data preprocessing steps such as handling missing values, encoding categorical variables, and feature scaling are performed to ensure the accuracy and robustness of the models. We evaluate model performance using metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC) to determine the best-performing model.

The results of the analysis show that specific factors, such as low job satisfaction, inadequate compensation, and lack of career growth opportunities, significantly contribute to employee churn.

This project highlights the importance of data-driven insights in employee retention. By predicting employee churn, organizations can take proactive measures to improve workplace conditions, enhance employee engagement, and reduce the cost of turnover, ultimately contributing to better long-term organizational performance.

TABLE OF CONTENTS

CHAPTER 1

Introduction

1.1 Problem Statement: Creating a prediction model for Employee Churn Prediction using Python.

1.2 Motivation: This Project was chosen as employee churn is a major problem in any company. This is a major issue that needs to be rectified as it can affect any company's standings.

1.3 Objective: Objectives of the project are predicting Employee Churn using employee data (Age,Salary,etc…) and creating a prediction model that can predict the Employee Churn and provide results to prevent that churn.

1.4 Scope of the Project: This project can provide a clear insight into the company's internal circumstances and the changes that need to be done to prevent the churn from happening.

CHAPTER 2

Existing Models

1. **Machine Learning Techniques for Employee Churn Prediction**

   A significant body of literature employs machine learning algorithms to predict employee churn. Early studies in employee attrition primarily relied on traditional statistical models, such as logistic regression and survival analysis. However, with advancements in machine learning (ML), more sophisticated models have been used, including decision trees, random forests, support vector machines (SVM), and neural networks.

- **Logistic Regression and Survival Analysis:** Studies by **Chien et al. (2010)** and **Feng et al. (2012)** demonstrate the use of logistic regression to predict turnover based on employee characteristics such as tenure, job satisfaction, and compensation. These models offered basic insights but lacked the flexibility of more complex ML techniques.

- **Decision Trees and Random Forests:** Research by **Guan et al. (2019)** and **Tziraki et al. (2021)** utilized decision trees and random forests, showing how these models could capture complex, nonlinear

relationships between employee characteristics and turnover risk. Random forests, in particular, are known for their ability to handle large datasets with many features and their ability to provide feature importance rankings, helping HR departments prioritize factors to improve retention.

- **Support Vector Machines (SVM): Burez and Van den Poel (2009)** applied SVM to predict employee churn and demonstrated that SVM, especially with a radial basis function (RBF) kernel, can outperform traditional methods by identifying subtle patterns in employee behavior.

- **Neural Networks:** Recent research, such as that by **Kaur et al. (2021)**, has explored the use of deep learning models for churn prediction. These models can learn from unstructured data (e.g., employee feedback, emails) and process large amounts of information with greater accuracy than simpler algorithms.

Limitations:

- **Imbalanced Data**: The number of employees who leave the organization is often much smaller than those who stay, making it harder to train effective models.
- **Feature selection**: Deciding which features are most predictive of churn can be difficult.

- **Data quality**: Incomplete or inconsistent employee data can reduce the accuracy of predictions.
- **Ethical concerns**: Predicting churn must be handled sensitively, as it can lead to biased decision-making or unfair treatment of employees.

CHAPTER 3

Proposed Methodology

3.1 System Design

This system design will include data engineering, model training, evaluation, deployment, and monitoring steps, ensuring scalability, accuracy, and interpretability.

The system design can be done by collecting data and analyzing the data so that the prediction can be done. The prediction model uses various classifiers and analyzers for this prediction models. For the project the model uses the following design after collecting data:

1. **Data Engineering Layer**
2. **Feature Store**
3. **Model Development Layer**
4. **Prediction Layer**

5. **Feedback and Monitoring Layer**
6. **User Interface and Visualization Layer**
7. **Data Privacy & Compliance**

Model

Exploratory Analysis

Exploratory Data Analysis is an initial process of analysis, in which you can summarize characteristics of data such as pattern, trends, outliers, and hypothesis testing using descriptive statistics and visualization.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Dataset

```
employee_df = pd.read_csv('Data Sets/Human_Resources.csv')
employee_df
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Employ
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	

1470 rows × 35 columns

employee_df.head(5)

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeN
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	

5 rows × 35 columns

employee_df.tail(10)

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Employ
1460	29	No	Travel_Rarely	468	Research & Development	28	4	Medical	1	
1461	50	Yes	Travel_Rarely	410	Sales	28	3	Marketing	1	
1462	39	No	Travel_Rarely	722	Sales	24	1	Marketing	1	
1463	31	No	Non-Travel	325	Research & Development	5	3	Medical	1	
1464	26	No	Travel_Rarely	1167	Sales	5	3	Other	1	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	

10 rows × 35 columns

```
employee_df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyR
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000

8 rows × 26 columns

Data Visualization

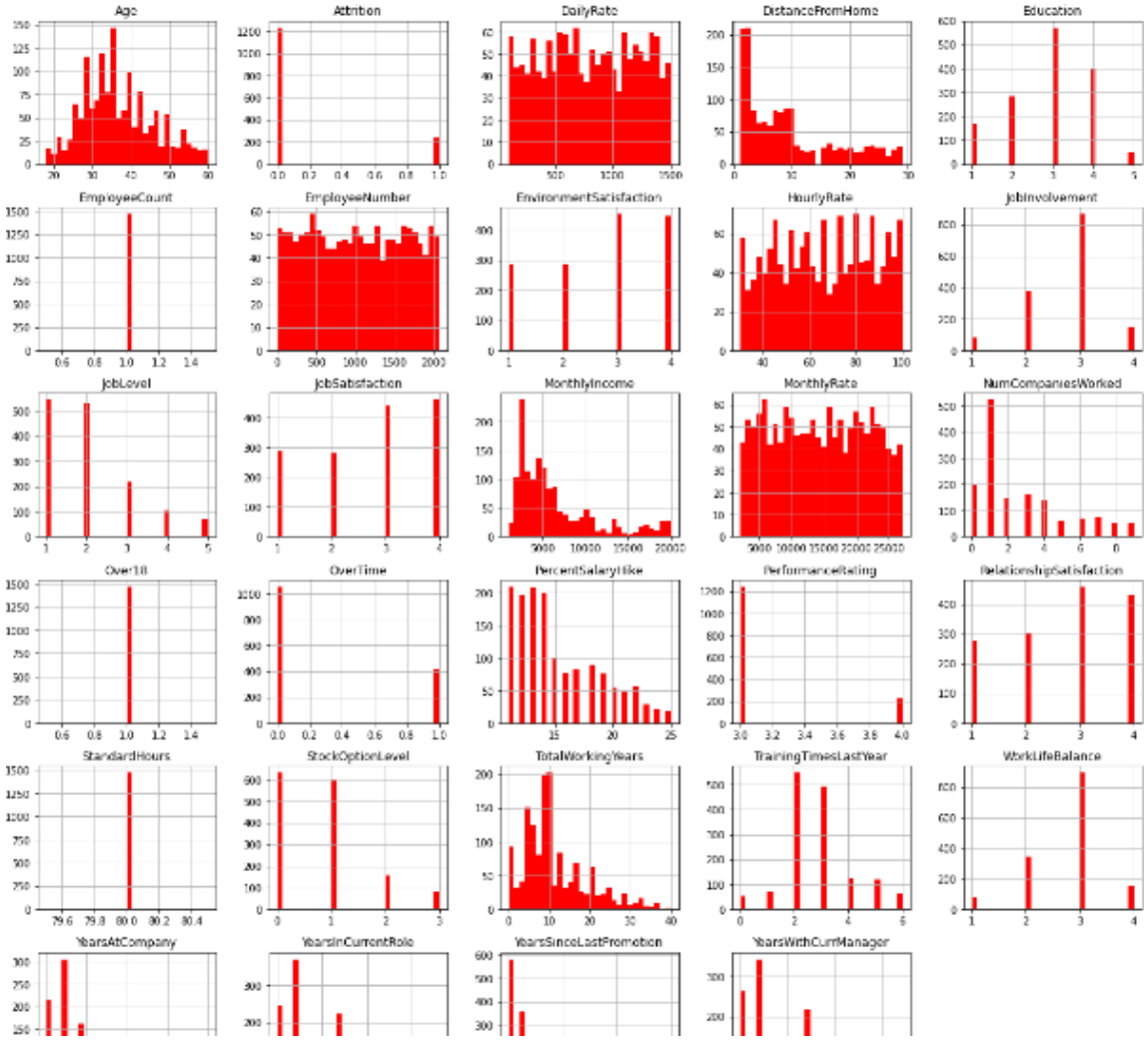
```
# Let's replace the 'Attritition' and 'overtime' column with integers before performing any visualizations

employee_df['Attrition'] = employee_df['Attrition'].apply(lambda x: 1 if x == 'Yes' else 0)
employee_df['OverTime'] = employee_df['OverTime'].apply(lambda x: 1 if x == 'Yes' else 0)
employee_df['Over18'] = employee_df['Over18'].apply(lambda x: 1 if x == 'Y' else 0)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	Employe
0	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	1	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	

4 rows × 35 columns

```
employee_df.hist(bins = 30, figsize = (20,20), color = 'r')
# Several features such as 'MonthlyIncome' and 'TotalWorkingYears' are tail heavy
# It makes sense to drop 'EmployeeCount' and 'Standardhours' since they do not change from one employee to the other
```



```
# It makes sense to drop 'EmployeeCount' , 'Standardhours' and 'Over18' since they do not change from one employee to the other
# Let's drop 'EmployeeNumber' as well
employee_df.drop(['EmployeeCount', 'StandardHours', 'Over18', 'EmployeeNumber'], axis=1, inplace=True)


# Let's see how many employees left the company!
left_df      = employee_df[employee_df['Attrition'] == 1]
stayed_df    = employee_df[employee_df['Attrition'] == 0]
# Count the number of employees who stayed and left
# It seems that we are dealing with an imbalanced dataset

print("Total =", len(employee_df))

print("Number of employees who left the company =", len(left_df))
print("Percentage of employees who left the company =", 1.*len(left_df)/len(employee_df)*100.0, "%")

print("Number of employees who did not leave the company (stayed) =", len(stayed_df))
print("Percentage of employees who did not leave the company (stayed) =", 1.*len(stayed_df)/len(employee_df)*100.0, "%")
```

```
Total = 1470
Number of employees who left the company = 237
Percentage of employees who left the company = 16.122448979591837 %
Number of employees who did not leave the company (stayed) = 1233
Percentage of employees who did not leave the company (stayed) = 83.87755102040816 %
```

```
left_df.describe()

# Let's compare the mean and std of the employees who stayed and left
# 'age': mean age of the employees who stayed is higher compared to who left
# 'DailyRate': Rate of employees who stayed is higher
# 'DistanceFromHome': Employees who stayed live closer to home
# 'EnvironmentSatisfaction' & 'JobSatisfaction': Employees who stayed are generally more satisifed with their jobs
# 'StockOptionLevel': Employees who stayed tend to have higher stock option level
```

	Age	Attrition	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	J
count	237.000000	237.0	237.000000	237.000000	237.000000	237.000000	237.000000	237.000000	237
mean	33.607595	1.0	750.362869	10.632911	2.839662	2.464135	65.573840	2.518987	1
std	9.689350	0.0	401.899519	8.452525	1.008244	1.169791	20.099958	0.773405	0
min	18.000000	1.0	103.000000	1.000000	1.000000	1.000000	31.000000	1.000000	1
25%	28.000000	1.0	408.000000	3.000000	2.000000	1.000000	50.000000	2.000000	1
50%	32.000000	1.0	699.000000	9.000000	3.000000	3.000000	66.000000	3.000000	1
75%	39.000000	1.0	1092.000000	17.000000	4.000000	4.000000	84.000000	3.000000	2
max	58.000000	1.0	1496.000000	29.000000	5.000000	4.000000	100.000000	4.000000	5

8 rows × 25 columns

```
stayed_df.describe()
```

	Age	Attrition	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement
count	1233.000000	1233.0	1233.000000	1233.000000	1233.000000	1233.000000	1233.000000	1233.000000
mean	37.561233	0.0	812.504461	8.915653	2.927007	2.771290	65.952149	2.770479
std	8.888360	0.0	403.208379	8.012633	1.027002	1.071132	20.380754	0.692050
min	18.000000	0.0	102.000000	1.000000	1.000000	1.000000	30.000000	1.000000
25%	31.000000	0.0	477.000000	2.000000	2.000000	2.000000	48.000000	2.000000
50%	36.000000	0.0	817.000000	7.000000	3.000000	3.000000	66.000000	3.000000
75%	43.000000	0.0	1176.000000	13.000000	4.000000	4.000000	83.000000	3.000000
max	60.000000	0.0	1499.000000	29.000000	5.000000	4.000000	100.000000	4.000000

8 rows × 25 columns

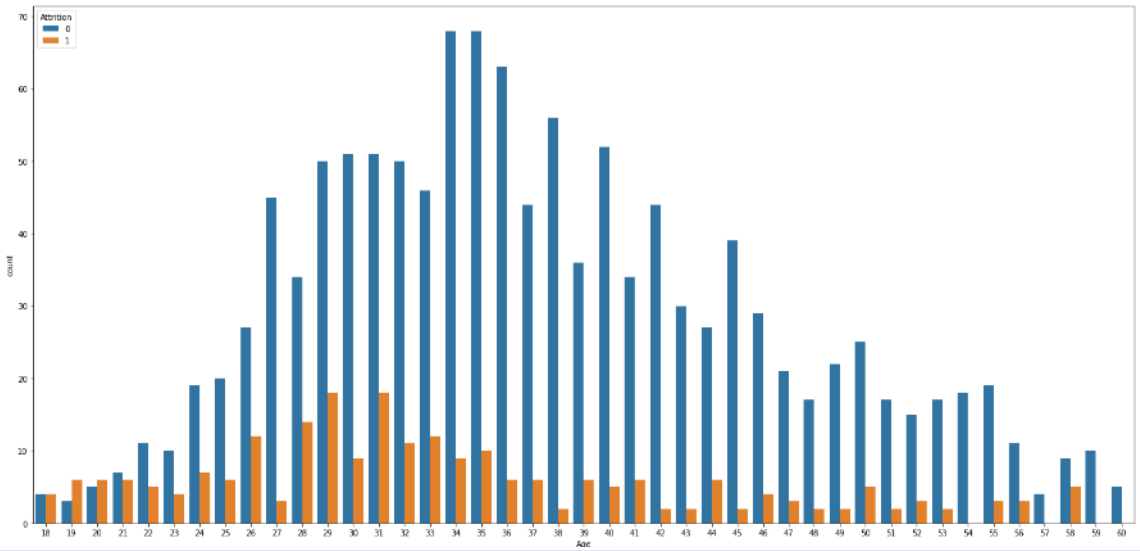
```
correlations = employee_df.corr()
f, ax = plt.subplots(figsize = (20, 20))
sns.heatmap(correlations, annot = True)

# Job level is strongly correlated with total working hours
# Monthly income is strongly correlated with Job level
# Monthly income is strongly correlated with total working hours
# Age is stongly correlated with monthly income
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x219fac27fd0>
```


```
plt.figure(figsize=[25, 12])
sns.countplot(x = 'Age', hue = 'Attrition', data = employee_df)
```

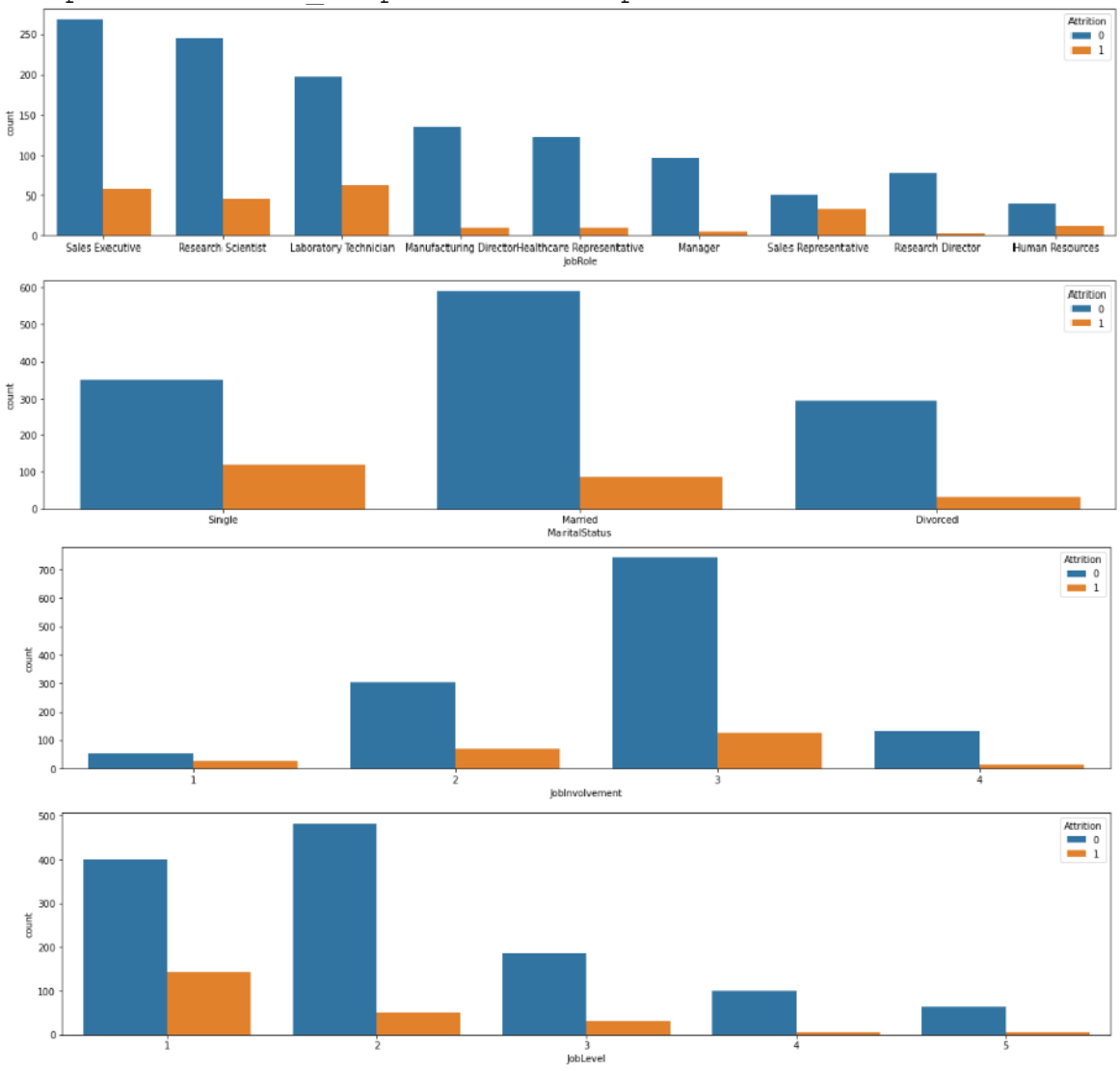
```
<matplotlib.axes._subplots.AxesSubplot at 0x219fb0e2ac0>
```




```
plt.figure(figsize=[20,20])
plt.subplot(411)
sns.countplot(x = 'JobRole', hue = 'Attrition', data = employee_df)
plt.subplot(412)
sns.countplot(x = 'MaritalStatus', hue = 'Attrition', data = employee_df)
plt.subplot(413)
sns.countplot(x = 'JobInvolvement', hue = 'Attrition', data = employee_df)
plt.subplot(414)
sns.countplot(x = 'JobLevel', hue = 'Attrition', data = employee_df)
```

```
# Single employees tend to leave compared to married and divorced
# Sales Representitives tend to leave compared to any other job
# Less involved employees tend to leave the company
# Less experienced (low job level) tend to leave the company
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x219fe147070>
```



Cluster Analysis:

Let's find out the groups of employees who left. You can observe that the most important factor for any employee to stay or leave is satisfaction and performance in the company. So let's bunch them in the group of people using cluster analysis.

```
#import module

from sklearn.cluster import KMeans

# Filter data

left_emp =  data[['age', 'salary']][data.left == 1]

# Create groups using K-means clustering.

kmeans = KMeans(n_clusters = 3, random_state = 0).fit(left_emp)
```

Building a Prediction Model

Pre-Processing Data

Lots of machine learning algorithms require numerical input data, so you need to represent categorical columns in a numerical column.

In order to encode this data, you could map each value to a number. e.g. Salary column's value can be represented as low:0, medium:1, and high:2.

This process is known as label encoding, and sklearn conveniently will do this for you using LabelEncoder.

```
employee_df.head(3)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction
0	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	2
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3
2	37	1	Travel_Rarely	1373	Research & Development	2	2	Other	4

3 rows × 31 columns

```
X_cat = employee_df[['BusinessTravel', 'Department', 'EducationField',
'Gender', 'JobRole', 'MaritalStatus']]
X_cat
```

	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus
0	Travel_Rarely	Sales	Life Sciences	Female	Sales Executive	Single
1	Travel_Frequently	Research & Development	Life Sciences	Male	Research Scientist	Married
2	Travel_Rarely	Research & Development	Other	Male	Laboratory Technician	Single
3	Travel_Frequently	Research & Development	Life Sciences	Female	Research Scientist	Married
4	Travel_Rarely	Research & Development	Medical	Male	Laboratory Technician	Married
...	...	...	...	...	...	...
1465	Travel_Frequently	Research & Development	Medical	Male	Laboratory Technician	Married
1466	Travel_Rarely	Research & Development	Medical	Male	Healthcare Representative	Married
1467	Travel_Rarely	Research & Development	Life Sciences	Male	Manufacturing Director	Married
1468	Travel_Frequently	Sales	Medical	Male	Sales Executive	Married
1469	Travel_Rarely	Research & Development	Medical	Male	Laboratory Technician	Married

1470 rows × 6 columns

```
from sklearn.preprocessing import OneHotEncoder
onehotencoder = OneHotEncoder()
X_cat = onehotencoder.fit_transform(X_cat).toarray()
X_cat.shape
```

```
(1470, 26)
```

```
X_cat = pd.DataFrame(X_cat)
X_cat
```

	0	1	2	3	4	5	6	7	8	9	...	16	17	18	19	20	21	22	23	24	25
0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
1	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
2	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
4	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1466	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1467	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1468	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
1469	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

1470 rows × 26 columns

```
# note that we dropped the target 'Atrittion'
X_numerical = employee_df[['Age', 'DailyRate', 'DistanceFromHome',
       'Education', 'EnvironmentSatisfaction', 'HourlyRate',
'JobInvolvement',       'JobLevel',       'JobSatisfaction',
       'MonthlyIncome',       'MonthlyRate',  'NumCompaniesWorked'
'OverTime',       'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction',       'StockOptionLevel',
       'TotalWorkingYears'    ,'TrainingTimesLastYear'        ,
'WorkLifeBalance',       'YearsAtCompany'       ,'YearsInCurrentRole',
'YearsSinceLastPromotion',       'YearsWithCurrManager']]
X_numerical
```

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfactio
0	41	1102	1	2	2	94	3	2	
1	49	279	8	1	3	61	2	2	
2	37	1373	2	2	4	92	2	1	
3	33	1392	3	4	4	56	3	1	
4	27	591	2	1	1	40	3	1	
...	...	...	...	...	...	...	...	...	
1465	36	884	23	2	3	41	4	2	
1466	39	613	6	1	4	42	2	3	
1467	27	155	4	3	2	87	4	2	
1468	49	1023	2	3	4	63	2	2	
1469	34	628	8	3	2	82	4	2	

1470 rows × 24 columns

```
X_all = pd.concat([X_cat, X_numerical], axis = 1)
X_all
```

	0	1	2	3	4	5	6	7	8	9	...	PerformanceRating	RelationshipSatisfaction	StockOptionLevel	TotalWorkingYe
0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	...	3	1	0	
1	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	4	4	1	
2	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	3	2	0	
3	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	3	3	0	
4	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	4	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	3	1	
1466	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	1	1	
1467	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	...	4	2	1	
1468	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	...	3	4	0	
1469	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	...	3	1	0	

1470 rows × 50 columns

3.2 Advantages

- This model can predict the Employee Churn based on the personal data of the Employees.
- As personal data is taken into account, the model accuracy increases compared to other existing models.
- This model can provide a insight into the employee working environment and what are the factors that affect them to leave the company.

3.3 Requirement Specification

3.5.1.Hardware Requirements:

Component	CPU	Memory	Storage	GPU	Network

Model Training	8-core+	32-128 GB	1 TB SSD	NVIDIA A100 or similar	10 Gbps
Batch Inference	4-8 cores	16-32 GB	500 GB SSD	Not required	1 Gbps
Real-Time Inference	4 cores, 3 GHz+	16-32 GB	256-500 GB SSD	NVIDIA T4 (optional)	1 Gbps
Data Storage	8-core+	32 GB	4-10 TB NAS/SAN	Not required	10 Gbps
Feature Store	4-8 cores per node	32 GB	Distributed SSD	Not required	High-speed, distributed
Monitoring & Logging	4-core	16 GB	500 GB - 1 TB SSD	Not required	1 Gbps

Software Requirements:

Category	Software
ETL and Data Processing	Apache Airflow, Apache Spark, Pandas
Feature Store	Tecton, Feast, Redis, Cassandra
Model Development	Scikit-learn, XGBoost, TensorFlow, PyTorch
Experiment Tracking	MLflow, Weights & Biases
Hyperparameter Tuning	Optuna, Hyperopt
Deployment	Docker, Kubernetes, SageMaker, FastAPI
Monitoring and Logging	Prometheus, Grafana, ELK Stack

User Interface	Tableau, Power BI, Streamlit, SHAP
Data Privacy & Security	Vault, IAM, TLS/SSL

CHAPTER 4

Implementation and Result

Evaluating Model Performance:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X = scaler.fit_transform(X_all)
X
```

```
array([[0.        , 0.        , 1.        , ..., 0.22222222, 0.        ,
        0.29411765],
       [0.        , 1.        , 0.        , ..., 0.38888889, 0.06666667,
        0.41176471],
       [0.        , 0.        , 1.        , ..., 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.        , 1.        , ..., 0.11111111, 0.        ,
        0.17647059],
       [0.        , 1.        , 0.        , ..., 0.33333333, 0.        ,
        0.47058824],
       [0.        , 0.        , 1.        , ..., 0.16666667, 0.06666667,
        0.11764706]])
```

```
y = employee_df['Attrition']
y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
1465    0
1466    0
1467    0
1468    0
1469    0
Name: Attrition, Length: 1470, dtype: int64
```

```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
# Model Precision
print("Precision:",metrics.precision_score(y_test, y_pred))
# Model Recall
print("Recall:",metrics.recall_score(y_test,  y_pred))
```

```
Accuracy: 0.971555555556
Precision: 0.958252427184
Recall: 0.920708955224
```

CHAPTER 5

Discussion and Conclusion

5.1 Key Findings:

This model predicts the Employee Churn based on the extensive data that is taken from Employee working environment.

This model analyses various factors that are involved in the Churn of the Employees. Some are as follows,

- **Job Involvement**
- **Education**
- **Job Satisfaction**
- **Performance Rating**
- **Relationship Satisfaction**
- **Work Life Balance**

5.2 **Git Hub Link of the Project:** https://github.com/Subashan-080/Employee-chrun-prediction.git

5.3 **Video Recording of Project** Demonstration:

5.4 **Limitations:**

- This model only analyses the factors based on the external data, but this data can differ based on circumstances.

- This model does not consider the unforeseen circumstances that happen to a employee.
- This model does not consider unethical concerns such as employee disagreement towards the prediction as employee may not leave the company but the prediction model says he will leave the company.

5.5 Future Work:

In future, the prediction model may also consider the unforeseen circumstances and also provide solutions for the unethical concerns.

5.6 Conclusion:

In conclusion, this model provides the overall internal standing of the company. But no model can provide 100% accurate results in Employees Churn. Thus predicting the Employee standings may provide the insights for the company and the HR to decide whether the company needs new employees or to improve the working environment for the employees.

REFERENCES

1. Research Papers

- **Predicting Employee Turnover Using Machine Learning**
  Homam Ghaffari, Zohreh Nasiripour, Mohammad Sadegh Aslani
  International Journal of Human Capital and Information Technology Professionals, 2019.
  This study explores different machine learning techniques for predicting employee churn and discusses factors affecting turnover, including job satisfaction and career development.

- **A Comprehensive Review of Employee Turnover Prediction Using Machine Learning**
  Aravind Manoharan, Amritha Dhinakaran, Mohamed Nabeel Mohamed Hussain
  2021 IEEE International Conference on Communication and Signal Processing (ICCSP).
  This paper provides a review of machine learning algorithms applied to employee turnover prediction, discussing feature engineering, model selection, and comparison of ML techniques.

2. Case Studies and Practical Guides

- **IBM HR Analytics Employee Attrition & Performance Dataset on Kaggle**
  A well-known dataset available on Kaggle, which includes various features related to employee turnover. This dataset has been widely used in tutorials and case studies for developing churn prediction models.

- **"Predict Employee Turnover with Python" - Towards Data Science (Medium)**
  This article provides a practical walkthrough of an employee churn prediction project using Python and machine learning. It covers data preprocessing, feature engineering, model training, and evaluation.

APPENDICES

Online Courses and Tutorials

- **Coursera: People Analytics by the University of Pennsylvania**
  This course provides a foundation in HR analytics and predictive modeling, including sections on employee turnover prediction.
- **Udacity: Predictive Analytics for Business**
  A course that covers predictive modeling techniques, including use cases for HR and churn prediction.
- **DataCamp: Machine Learning for Time Series Data in Python**
  DataCamp offers specific modules on time-series prediction, which can be helpful if you incorporate historical trends or time-based features in churn prediction.