PEOFARMA FOR THE APPROVAL PROJECT PROPOSAL

PRN NO. **ROLL NO.**

1. Name of the Student

Hemangi Eknath Mestry.

2. Title of the Project

HR Attrition and Performance

3. Name of the Guide

Assi. Prof. Ninad Dani Sir.

4. Teaching experience of the guide

5. Is this your first submission? **Yes** **No**

Signature of the Student **Signature of the Guide**

Date: ……. **Date: …….**

HR ATTRITION AND PERFORMANCE WEB APPLICATION DASHBOARD

A PROJECT REPORT

Submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF SCIENCE INFORMATION TECHNOLOGY

By

Miss. HEMANGI EKNATH MESTRY.

SEAT NO.:

Under the esteemed guidance of

Asst. Prof. Mr. NINAD DANI SIR.

DEPARTMENT OF INFORMATION TECHNOLOGY

VIJAYALAKSHMI VISHWANATH DALVIE COLLEGE TALERE

(UGC & Govt. of Maharashtra Funded Model College)

TALERE, TAL-KANKAVLI, DIST-SINDHUDURG, PIN-416801
MAHARASHTRA

2023-24

VIJAYALAKSHMI VISHWANATH DALVIE COLLEGE TALERE

(UGC & Govt. of Maharashtra Funded Model College)

TALERE, TAL- KANKAVLI, DIST-SINDHUDURG, PIN-416802

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, “HR Attrition and Performance”, is bonafied work of Miss. Hemangi Eknath Mestry. Bearing Seat No. submitted in partial fulfilment of the requirements of the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Co-ordinator**

External Examiner

Date: …… **College Seal**

ABSTRACT

Today's world, every company is interested in predicting employee performance and attrition based on employee performance. This Dissertation aims to help the HR and Project Managers in improving the retention rate of the valuable employees in an organization, thereby minimizing the employee turnover cost of the company.

The Project was carried out in three stages. To improve the retention rate, efforts were made to first, predict the employee attrition; secondly, decide on which employees are valuable and their retention is profitable to the company. Finally, the factors that influence the employee's intention to resign from a company is found out and provided to the HR and Project managers through the HR Analytical application developed using Streamlit framework. Also, the survey for these factors has also been carried out among the HR professionals. Decision Model was prepared with the help of conditional logic statements which showcased which employees are valuable and which employees are not. Then the separate employee source file was prepared consisting of the valuable employees and who were also a potential candidate for resignation. The dashboard was developed which shows all the factors influencing employee attrition so that HR and Project Managers can use them accordingly retaining valuable employee. For developing and testing the application historical employee dataset was used. This application can be used by the HR Managers to simplify the employee retention decision.

ACKNOWLEDGEMENT

We express our heartfelt gratitude to all those who contributed to the successful completion the project. This project would not have been possible without the support, guidance, and encouragement of various individuals and institutions.

We would like to express our deep gratitude to our honourable Principal for giving us an opportunity to study and pursue a career in Information Technology.

I would like to express my appreciation to our Head of the department and my project guide Mr. Ninad Dani Sir for her valuable and constructive suggestions during the planning and development of this project work.

A special thanks to all our professors of Information Technology for their valuable advises at every stage of work. Also, we are extending our thanks to lab assistants and admins for their seemingly small but valuable help for timely Internet Access and Lab access.

Our gratitude extends to our peers and classmates for their encouragement and support.

Last but not least, we are indebted to our families and friends for their unwavering encouragement and understanding throughout the duration of this project.

DECLARATION

I here by declare that the project entitled, “HR Attrition and Performance Web Application Dashboard” done at Talere, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as a part of our curriculum.

Name and Signature of the Student

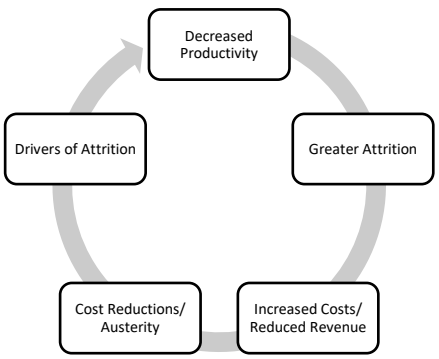
Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Background

Attrition rate is characterized as the number of employees who can leave an organization during a predefined time frame isolated by the total number of employees throughout the equivalent time span. It is costly non gainful and baffling. Harvard Business review article assessed the expense of attrition at over 150 percent of an employee's yearly compensation and could influence 40 percent of organization benefits.



Employee Attrition Cycle

The above mention diagrams show the cycle of employee attrition

- Attrition reduces productivity
- People leaving an organization increase pressure on other employees.
- This contributes to more employee to leave the company, which further cost more costs and less revenue.
- This forces the organization to take preventive measures for additional cost reduction and austerity measures.
- This causes working environment more difficult, causes the best employees with the most external opportunities to leave the company.

Employee attrition is a major problem for organization, ranges between 15% and 20%. An association does not face any consequences because of certain person leaving an organization. The company experiences the consequences because the employee

leaving the company takes away certain knowledge, and there is no ready replacement in the market.

1.2 Objectives

1. The objective of this project is to perform an Exploratory Data Analysis (EDA) and Statistical Data Analysis on the HR Analytics Employee Attrition dataset using Python.

2. The goal is to gain insights into the factors influencing employee attrition and performance within the organization.

3. By examining various features and conducting statistical analysis, we aim to identify patterns, trends, and potential areas for improvement that can help the company better understand and manage employee attrition and performance.

4. Document the entire process, including data collection, analysis methodologies, findings, and conclusions. Prepare a comprehensive project report and presentation.

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose:

The purpose of the "HR Attrition and Performance" project is to utilize data analytics methodologies to gain deeper insights into employee attrition and performance within organizations. The project aims to apply data-driven approaches to address the challenges of retaining talent and optimizing employee productivity. By analysing historical HR data and employing advanced analytics techniques, the project seeks to provide HR professionals with actionable insights and recommendations that can contribute to effective talent management strategies, enhanced employee retention, and improved overall organizational performance. Through this project, students will develop practical skills in data analysis and gain a better understanding of how data-driven decision-making can significantly impact human resource management practices.

1.3.2 Scope:

The scope of an HR attrition and performance web application dashboard typically includes a range of features and functionalities designed to assist HR professionals in managing employee attrition and monitoring performance. Here are some common components and aspects that might be included in such a dashboard:

1. Employee Data Management:

- The dashboard should allow HR to input the Dataset of company and maintain employee data, including Department, job details, and performance metrics.

2. Performance Metrics:

- Display key performance indicators (KPIs) for the entire organization.

- Include metrics such as Overall Employees, Attrition Rate, Average Performance

3. Reporting and Analytics:

- Generate various reports and analytics to provide insights into attrition and performance trends.

4. User-Friendly Interface:

- Ensure the dashboard is easy to navigate and provides an intuitive user experience for HR personnel.

5. Mobile Accessibility:

- Consider making the dashboard accessible via mobile devices for remote HR management.

The scope of the dashboard can vary depending on the organization's specific needs and priorities. Some organizations may focus more on attrition prediction and retention strategies, while others may prioritize performance management and talent development. The key is to tailor the dashboard to meet the unique requirements of the HR department and support informed decision-making.

1.3.3 Applicability:

This project is applicable to a wide range of organization across various industries.

It is particularly relevant to:

1. HR Professional:

- Equipping HR practitioners with data-driven insights to effectively manage talent, reduce attrition rates, and optimize employee performance.

2. Organisational Decision-Makers:

- Providing senior management with evidence-based recommendations to make informed decisions about HR strategies and resource allocation.

3. Academic Institution:

- Serving as a learning tool for students in Information technology, data science, and human resources disciplines to bridge theoretical knowledge with practical application.

CHAPTER 2

SURVEY OF TECHNOLOGIES

I am using Data Mining Techniques and python language for developing the HR attrition and Performance Dashboard Web application.

2.1 Why to use python?

Python suits a variety of web project, from simple to complex. Python language is Ease of learning and use, Vast Ecosystem of libraries and frameworks, Data analysis and Visualization, Machine learning Capabilities, Cross-Platform Compatibility, Rapid prototyping and development and Open Source and Cost-Effective.

2.2 Stream-lit Library

Stream-lit is an open-source Python library that enables developers to create interactive web applications for data science and machine learning projects. It allows you to convert data scripts into shareable web apps with minimal effort, making it an excellent tool for building data-driven dashboard, visualizations, and user interfaces. Stream-lit simplifies the process of creating web applications by eliminating the need to HTML, CSS or JavaScript code. Instead, you write Python script that define the app's layout, interactivity and presentation.

2.3 Data Visualization Libraries

1. pandas:

   It's used for data manipulation and analysis in HR attrition and performance web application dashboards, enabling data handling, visualization, and analysis of employee-related metrics efficiently.

2. streamlit:

   It simplifies development by allowing data visualization, user-friendly interfaces, and real-time updates, enhancing the user experience and facilitating data-driven decision-making for HR professionals.

3. numpy:

numpy is a Python library that's super handy for numerical computing. It gives you tools to work with arrays (kind of like lists, but more powerful) and perform mathematical operations on them efficiently. Numpy is great for tasks like matrix operations, statistical analysis, and generating random numbers. It's widely used in scientific computing, data analysis, and machine learning because it's fast and easy to use.

4. seaborn:

Seaborn is a Python library used for making attractive and informative statistical graphics. It builds on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics. It's great for visualizing data and relationships between variables, making it easier to understand and analyze datasets.

5. matplotlib.pyplot:

`matplotlib.pyplot` is a Python library used to create visualizations like graphs, charts, and plots. With `pyplot`, you can customize your plots with various styles, colors, labels, and more, making your data presentation visually appealing and informative. It's widely used in data analysis, scientific research, and other fields where data visualization is essential.

6. scipy:

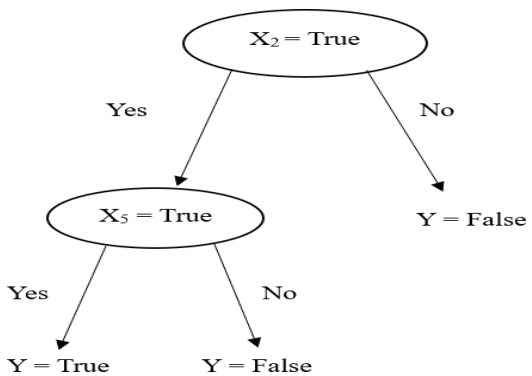
Scipy is a collection of scientific computing tools in Python. It includes modules for optimization, integration, interpolation, linear algebra, statistics, and more. It's like a Swiss Army knife for scientists and engineers, providing ready-to-use functions for complex mathematical tasks. Whether you're crunching numbers, fitting curves, or solving differential equations, Scipy has got you covered, making advanced scientific computing accessible to Python users.

Data Mining Algorithm Used:

DECISION TREE

Decision Tree is a type of supervised machine learning algorithm which is used to build regression and classical model. It uses a tree like model to make decision and possible results. It is one of the famous tools in machine learning. This technique was published first by Sonquist and Morgan in 1960s. Decision tree is also used in data mining.

Decision trees are generally used because it is conceptually easy but still a powerful machine learning model. It is very much capable to handle the missing values and independent features and it can select variables automatically.



Advantages of Decision Tree

- It does not require scaling of data.
- Needs less effort for the data processing and preparation.
- Missing values does not affect the result.

Disadvantages of Decision Tree

- Requires more time for training the model.
- A small edit in data can cause a huge deviation in results.
- Time complexity is more as compared to other models.
- It is inadequate for continuous values.

CHAPTER 3

REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

The problem definition for a project focused on HR attrition and performance involves outlining the specific challenges and goals you aim to address using data-driven insights.

In today's dynamic and competitive business landscape, organizations face significant challenges in managing employee attrition and optimizing performance. High attrition rates can lead to increased recruitment and training costs, reduced team morale, and potential disruptions in productivity. Moreover, understanding and enhancing employee performance is crucial for maintaining a motivated and efficient workforce.

The HR department of our organization aims to tackle these challenges by leveraging data analytics techniques to comprehensively analyse attrition patterns and employee performance. The objective is to identify the underlying factors contributing to attrition, predict attrition risks, and develop effective retention strategies. Additionally, the organization seeks to gain insights into performance trends, factors influencing performance outcomes, and areas where targeted interventions can lead to improvements.

By employing data analytics, we intend to answer critical questions such as:

- What are the primary drivers of employee attrition within the organization?

- Can we predict attrition risk for different employee segments, allowing for proactive retention efforts?

- How do various factors, such as job role, job satisfaction, and career growth, impact attrition rates?

- What are the characteristics of high-performing employees, and how can we replicate those qualities in others?

- Are there correlations between employee engagement, training initiatives, and overall performance?

- How can we design tailored interventions to enhance both employee retention and performance?

Through this project, we aim to provide the HR department and organizational leadership with actionable insights to make informed decisions, improve HR strategies, reduce attrition rates, and foster a more engaged and high-performing workforce. By addressing these challenges, we strive to contribute to the long-term success and sustainability of the organization.

3.2 Requirements Specification

***Functional Requirement*:**

Data collection: Collect relevant data from Kaggle website.

Employee Data Display: Display employee Gender, Age, Department, Role, Performance, Attrition, etc.

Attrition Analysis: Visualize attrition trends, reasons for attrition, and related metrics.

Data Visualization: Provide graphs, charts, and visualization to represent attrition and performance data effectively.

Interactive User Interface: A user-friendly interface for easy navigation and interaction.

***Key Objective*:**

Attrition Analysis: Develop a comprehensive understanding of employee attrition by exploring factors such as age, department, role and performance history.

Trend Visualization: Create visualization that offer clear insights into attrition trends, performance distribution, and correlation between various factors.

3.3 Planning and Scheduling

For planning and scheduling for my Dashboard I chose Agile model.

AGILE MODEL:

I chose Agile Methodology for developing my project HR Attrition and Performance web application due to its great approach. Agile methodology emphasizes iterative development, allowing user adapt to changing requirement and refine to over time. In this project create user stories for features like attrition prediction, performance analysis, EDA (Exploratory Data Analysis) and reporting. Regular feedback from HR manager will help shape the app's functionality to better meet the needs of HR professionals.

Agile Model has following phases:

1. Requirements Gathering
2. Planning
3. Design
4. Coding
5. Testing
6. Implementation
7. Maintenance

<u>1. Requirement Gathering:</u>

i. In Requirement gathering phase I will conduct the meeting with HR manager to get requirement for my web application. To analyse need for an organization to assess valuable employees to maintain growth and stability of an organization.

<u>2. Planning:</u>

i. In planning I am use Gantt chart for scheduling the project.
ii. In the planning phase I will create Road-map about how to full-fill requirement of the HR manager.
   - Collect the Data from the HR manager.

- Find the necessary hardware and software requirements for the project.
- Study the algorithm which are needed for developing project.
- Create a rough template for User Interface.

3. Design:

i. The design phase includes front-end design and back-end design.
ii. Front-end design like user interface of the web application. In front-end design we use python language. The layout of my web app is look like as follows. I will draw rough UI to show the layout of my project.
iii. To understand the backend design I use some diagrams, include:
- UML: State Transition Diagram, Sequence Diagram, Use Case Diagram

4. Coding

i. As we know there are two types of coding first frontend coding and second one is backend coding.
ii. In this website I'm use Python for both the frontend and backend language. Stream-lit itself handles the communication between the frontend (browser) and backend (Python scripts) seamlessly.

5. Testing

i. Testing helps to counter the working of sub-assemblies, components, assembly, and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fail abruptly.

❖ The different ways of testing
➢ Functional Testing:

The type of testing facilitates in presenting the systematic representation that capabilities tested are to be had and specified with the aid of technical requirements, documentation of the system and the user manual.

- Regression Testing:

  As new features are added or changes are made to the application, regression testing verifies that existing functionality hasn't been negatively impacted. Automated regression testing can be particularly valuable in Agile development to quickly catch regression.

- Accessibility Testing:

  I will perform this type of testing to confirm that the application is accessible to user with disabilities.

6. Implementation:

i. Data Collection and Preparation:

- Gather relevant HR data, including employee information, performance metrics, attrition rates, etc.

- Clean and preprocess the data to ensure accuracy and consistency.

ii. Install Stream-lit and Required Libraries:

- Install Stream-lit using 'pip install stream-lit'.

- Import other necessary libraries like Pandas, NumPy, Matplotlib (for data visualization) etc.

iii. Data Visualization:

- Integrate data visualization libraries like Matplotlib or Plotly.express to create meaningful charts and graphs.

- Display visualizations that provide insights into attrition trends, performance metrics, employee distribution, etc.

iv. Dynamic Content Generation:

- Use Stream-lit features to dynamically generate content based on user interactions.

- Update visualizations and data display as users select different filters or options.

v. Historical Data and Trends:

- Include visualizations that show historical data and trends over time.

- Use line charts, bar charts, or heatmaps to display changes in attrition and performance metrics.

vi. User-Friendly UI:

- Design a user-friendly interface that guides users through the dashboard.

- Use well-organized sections, headings, and clear instructions.

vii. Responsive Design:

- Ensure your dashboard is responsive and accessible across different screen sizes and devices.

viii. Documentation and Deployment:

- Provide documentation on how to use the dashboard.

- Deploy the Stream-lit application to a web server or cloud platform for accessibility.

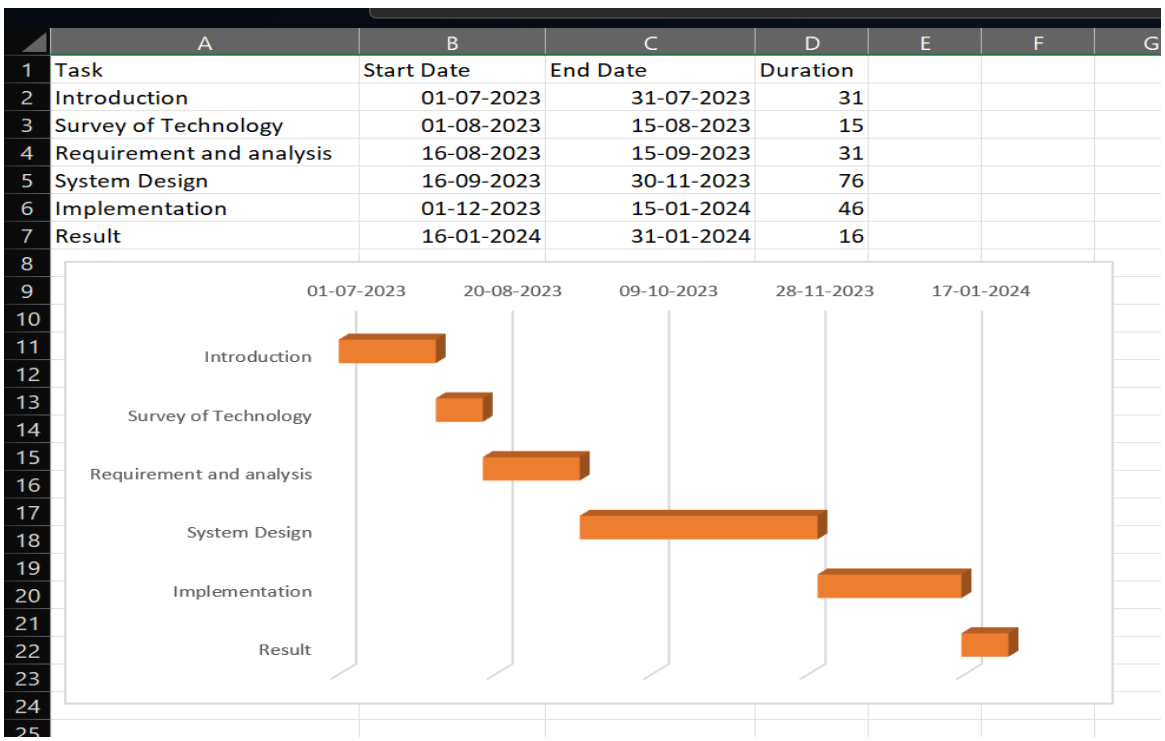
7. Maintenance:

i. Maintenance is essential to ensure its accuracy, functionality and usability over time.

- Bug Fixes: Continuously monitor the dashboard for any bugs or error that might arise from changes in the data or the system.

- User Feedback: Gather feedback from users about their experience with the dashboard.

- ***Gantt Chart***



3.4 Software and Hardware Requirements

Some of the hardware and software required for converting the abstract system into real one. The hardware and Software will require to develop my project that are as follows:

1. <u>Hardware Requirement</u>:

- A computer with intel i3 processor and 8GB RAM to handle the application and data processing.

2. Software Requirement:

- Operating System: Compatible with Windows 11.
- Python Language: The Project will be developed using Python programming language with Streamlit Framwork.
- Integrated Development Environment (IDE): Python IDLE.
- Required Python libraries: Stream-lit, Pandas, NumPy, Matplotlib, stream-lit-KPI.

3.5 Dataset Overview

Data collection is a process to collect relevant data from all sources possible for analysis. The dataset used for the prediction of employee turnover in which is obtained from Kaggle. This dataset consists of 35 features and 1470 rows. All the categorical data in each column were converted to numerical values by creating dummy columns.

For example., Job-Role values, which were either Sales Executive, Manager etc were converted to columns named Job-Role-Sales Executive, Job-Role-Manager and so on with values 0 or 1 so as to make them numerical data. The below diagram gives an overview of the dataset which is used in this project.

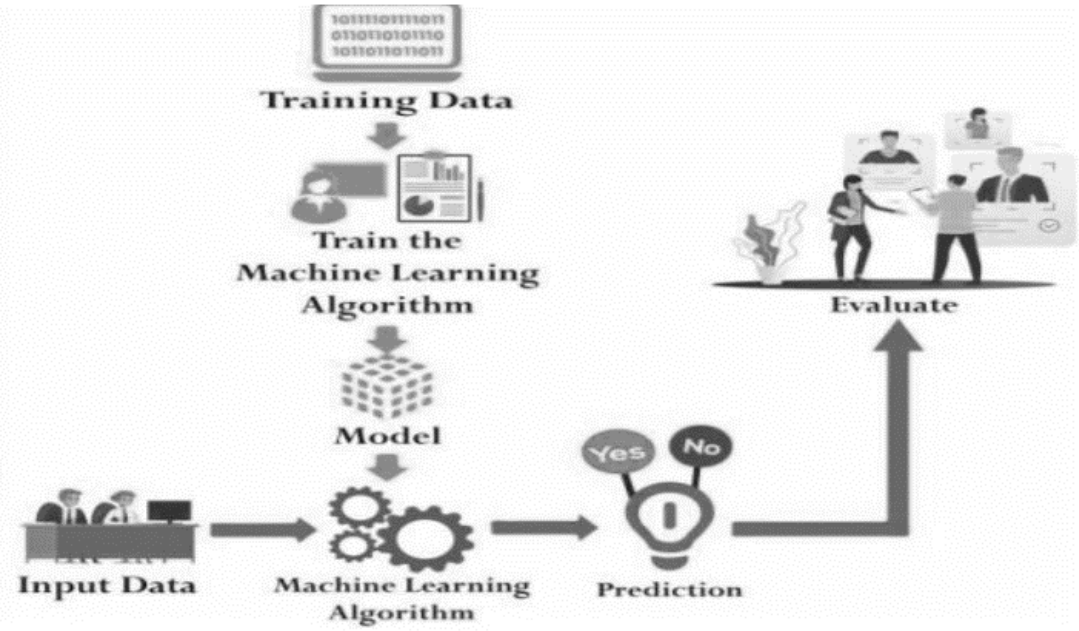
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relations
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows × 35 columns

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture Diagram



4.2 Basic Modules

HR Attrition and Performance Dashboard application is developed for HR Manager for Company. They will check Why Company Employee's Leave their organization? and What reason behind them? In this Application we use following Modules:

- User Module
- Metadata
- Data Processing Module

Modules Description:

User Module:

HR Manager is the user of this application, User upload the company's employee data and check why employees leave the organization with the help of Exploratory Data Analysis and Display various graphs and charts.

Users can interact with the dashboard to view HR metrics, generate reports and use filters to analyse data.

Metadata Module:

It manages the metadata that defines the structure and meaning of the data. This includes defining the various HR metrics, their sources and their relationship. It allows users and administrators to customize the dashboard layout and select which HR metrics they want to display.

Data Processing Module:

This module is responsible for extracting data from various sources such as performance record and surveys.

It transforms raw data into suitable format for analysis and visualization including aggregating, cleaning and merging data from different sources. It processes and visualizes the data; creating charts, graphs and tables for the dashboard.

4.3 UML Diagram

UML is simply anther graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

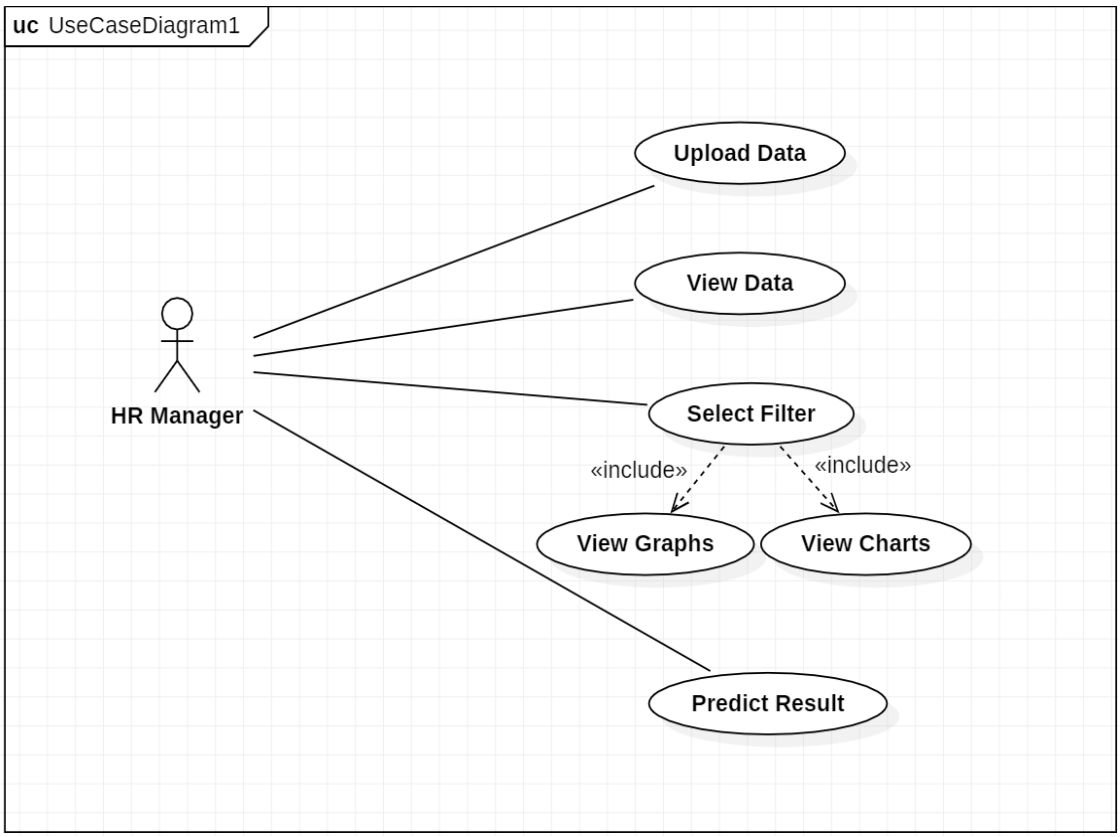
Advantages:

- To represent complete systems using object-oriented concepts.
- To establish an explicit coupling between concepts and executable code.
- To take into account the scaling factors that are inherent to complex.
- Critical end.
- To creating a modelling language usable by both humans and machines.

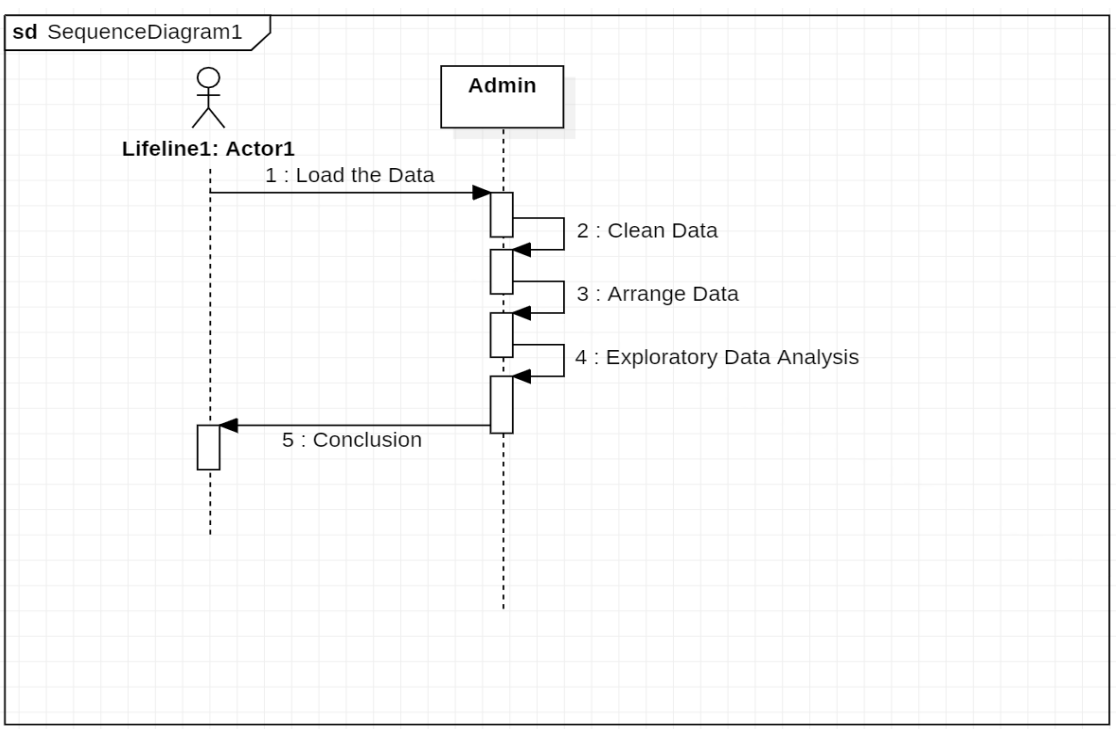
UML defines several models for representing systems:

- The class model captures the static structure
- The state model expresses the dynamic behaviour of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process.

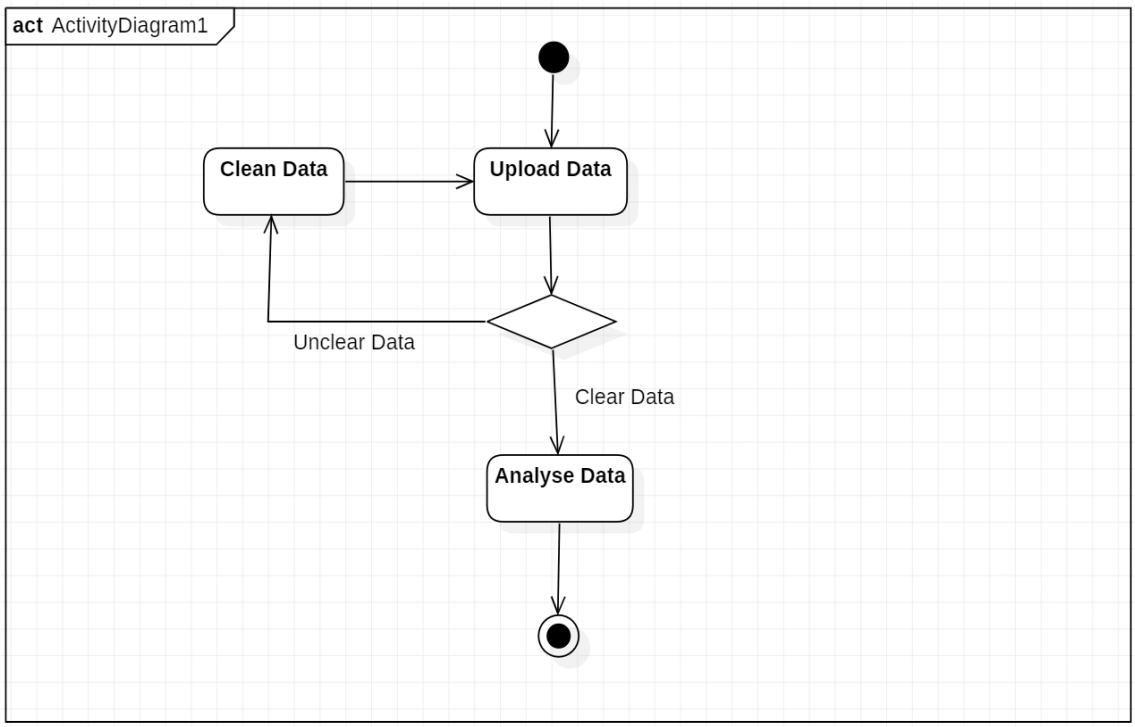
1) Use Case Diagram



2) Sequence Diagram



3) Activity Diagram



CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Code

Application > HR.py > ...

```
#Code for Visualization
st.set_page_config(layout="wide", page_icon=":bar_chart:", page_title='HR Attrition and Performance Dashboard')
st.image('HR1.png', width=150)
st.title("HR Attrition and Performance")

functions.space()
st.write('<p style="font-size:130%">Import Dataset</p>', unsafe_allow_html=True)

#Selecting CSV or Exel File
file_format = st.radio("Select File Format: ", ('csv','excel'), key='file_format')
dataset = st.file_uploader(label='')

#Default dataset
use_defo = st.checkbox('Use Example Dataset')
if use_defo:
    dataset = 'raw_hr_data1.csv'

#sidebar
st.sidebar.header('Import Dataset to Use Available Features: 👉')

if dataset:
    all_visuals = ['Dataset', 'Exploratory Data Analysis', 'Statistical Analysis']
    #functions.sidebar_space(2)
    visuals = st.sidebar.multiselect("Choose which visualizations you want to see 👇", all_visuals)
```

Application > HR.py > ...

```
    if 'Dataset' in visuals :
        if file_format == 'csv' or use_defo:
            df = pd.read_csv(dataset)
        else:
            df = pd.read_excel(dataset)

        st.header('_DataSet:_')
        n, m = df.shape
        st.write(f'<p style="font-size:130%">Dataset contains {n} rows and {m} columns.</p>', unsafe_allow_html=True)
        st.dataframe(df)

    #Exploratory Data Analysis
    if 'Exploratory Data Analysis' in visuals:
        all_visual=['Rate wise Employee Attrition','Gender wise Employee Attrition','Age wise Employee Attrition',
                    'Business Travel wise Employee Attrition','Department wise Employee Attrition','DailyRate wise Employe
                    'Distance From Home wise Employee Attrition','Education wise Employee Attrition',
                    'Education Field wise Employee Attrition','Environment Satisfaction wise Employee Attrition',
                    'JobRoles wise Employee Attrition','Job Level wise Employee Attrition',
                    'Job Satisfaction wise Employee Attrition','Marital Status wise Employee Attrition',
                    'Monthly Income wise Employee Attrition','Monthly Rate wise Employee Attrition',
                    'Number of Companies Worked wise Employee Attrition','Over Time wise Employee Attrition',
                    'Percentage Salary Hike wise Employee Attrition','Performance Rating wise Employee Attrition',
                    'Relationship Satisfaction wise Employee Attrition','Work Life Balance wise Employee Attrition',
                    'Total Working Years wise Employee Attrition','Years at Company wise Employee Attrition',
                    'Years In Current Role wise Employee Attrition','Years Since Last Promotion wise Employee Attrition',
```

Application > HR.py > ...

```
visual = st.sidebar.multiselect("Choose which features you want to explore on data 👇", all_visual)

# Visualize the Employee Attrition in Counts
st.header("_Exploratory Data Analysis:")

if 'Rate wise Employee Attrition' in visual:
    st.subheader("Employee Distribution by Attrition")
    if "Attrition" in df.columns:
        # Visualization to show Employee Attrition in Counts.
        plt.figure(figsize=(17, 6))
        plt.subplot(1, 2, 1)
        attrition_rate = df["Attrition"].value_counts()
        sns.barplot(x=attrition_rate.index, y=attrition_rate.values, palette=["#11f9f9", "#ffff00"])
        plt.title("Employee Attrition Counts", fontweight="black", size=20, pad=20)
        for i, v in enumerate(attrition_rate.values):
            plt.text(i, v, v, ha="center", fontweight='black', fontsize=18)

        # Visualization to show Employee Attrition in Percentage.
        plt.subplot(1, 2, 2)
        plt.pie(attrition_rate, labels=["No", "Yes"], autopct="%.2f%%", textprops={"fontweight": "black",
            "size": 15},colors=["#11f9f9", "#ffff00"], explode=[0, 0.1], startangle=90)
        center_circle = plt.Circle((0, 0), 0.3, fc='white')
        fig = plt.gcf()
        fig.gca().add_artist(center_circle)
        plt.title("Employee Attrition Rate", fontweight="black", size=20, pad=10)
        # Display the plots using Streamlit
        plt.tight_layout()
        st.pyplot(plt)
    else:
        st.error("The Uploaded Dataset Doesn't contain 'Attrition' columns.")
```

Application > HR.py > ...

```
#Visualization to show Total Employees by Department.
if 'Department wise Employee Attrition' in visual:
    if "Attrition" in df.columns and "Department" in df.columns:
        # Visualization to show Total Employees by Department.
        st.subheader("Employees Distribution by Department")
        plt.figure(figsize=(14, 6))
        plt.subplot(1, 2, 1)
        value_1 = df["Department"].value_counts()
        sns.barplot(x=value_1.index, y=value_1.values, palette=["#f67b4a", "#ea9d7c", "#FFC0CB"])
        plt.title("Employees by Department", fontweight="black", size=20, pad=20)
        for index, value in enumerate(value_1.values):
            plt.text(index, value, str(value), ha="center", va="bottom", fontweight="black", size=15)

        # Visualization to show Employee Attrition Rate by Department.
        plt.subplot(1, 2, 2)
        new_df = df[df["Attrition"] == "Yes"]
        value_2 = new_df["Department"].value_counts()
        attrition_rate = np.floor((value_2 / value_1) * 100).values
        sns.barplot(x=value_2.index, y=value_2.values, palette=["#660b30", "#dc2d73", "#f070a3"])
        plt.title("Attrition Rate by Department", fontweight="black", size=20, pad=20)
        for index, value in enumerate(value_2):
            plt.text(index, value, str(value) + " (" + str(attrition_rate[index]) + "% )", ha="center",
                va="bottom", size=13, fontweight="black")
        plt.tight_layout()
        # Display the plots using Streamlit
        st.pyplot(plt)
    else:
        st.error("The Uploaded Dataset Doesn't contain 'Department' columns.")
```

```
if 'Monthly Income wise Employee Attrition' in visual:
    st.subheader("Employee Distribution by Monthly Income")
    if 'MonthlyIncome' in df.columns:
        #Visualization to show Employee Distribution by MonthlyIncome.
        plt.figure(figsize=(13,6))
        plt.subplot(1,2,1)
        sns.histplot(x="MonthlyIncome", hue="Attrition", kde=True ,data=df,palette=["#11264e","#6faea4"])
        plt.title("Employee Attrition by Monthly Income",fontweight="black",size=20,pad=15)

        #Visualization to show Employee Attrition by Monthly Income.
        plt.subplot(1,2,2)
        sns.boxplot(x="Attrition",y="MonthlyIncome",data=df,palette=["#D4A1E7","#6faea4"])
        plt.title("Employee Attrition by Monthly Income",fontweight="black",size=20,pad=15)
        plt.tight_layout()
        st.pyplot(plt)
    else:
        st.error("The Uploaded Dataset Doesn't contain 'MonthlyIncome' columns.")
```

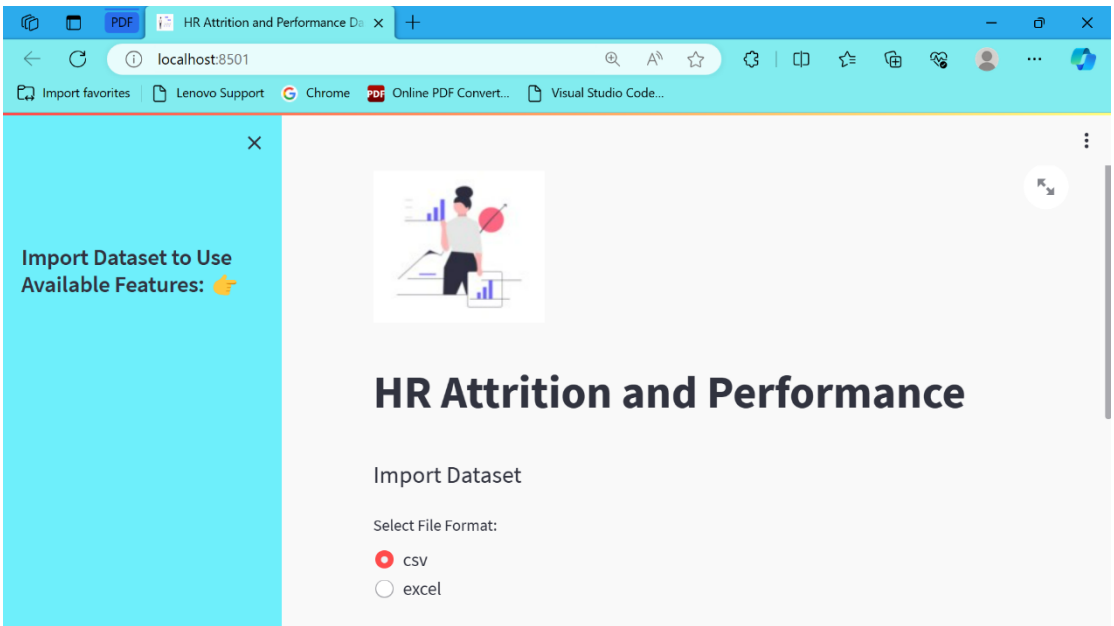
Application > HR.py > ...

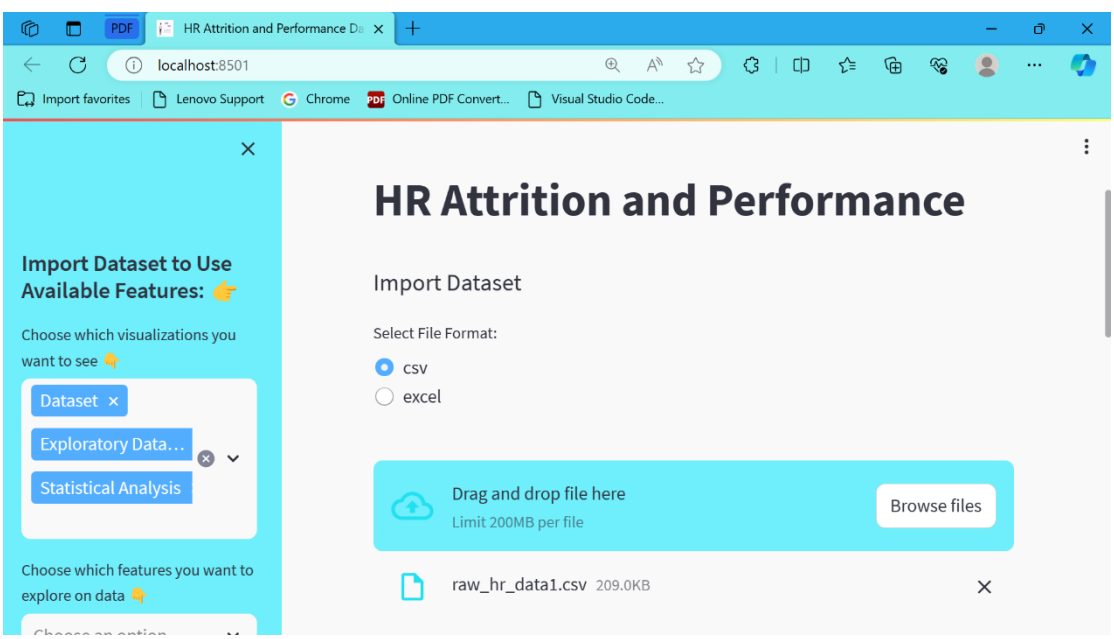
```
if 'Percentage Salary Hike wise Employee Attrition' in visual:
    st.subheader("Employee Distribution by Percentage Salary Hike")
    if 'PercentSalaryHike' in df.columns:
        #Visualization to show Employee Distribution by Percentage Salary Hike.
        plt.figure(figsize=(16,6))
        sns.countplot(x="PercentSalaryHike", hue="Attrition", data=df, palette=["#1d7874","#AC1F29"])
        plt.title("Employee Attrition By PercentSalaryHike",fontweight="black",size=20,pad=15)
        st.pyplot(plt)
    else:
        st.error("The Uploaded Dataset Doesn't contain 'PercentSalaryHike' columns.")

if 'Performance Rating wise Employee Attrition' in visual:
    st.subheader("Employee Distribution by Performance Rating")
    if 'PerformanceRating' in df.columns:
        df["PerformanceRating"] = df["PerformanceRating"].replace({1:"Low",2:"Good",3:"Excellent",4:"Outstanding"}
        #Visualization to show Total Employees by PerformanceRating.
        plt.figure(figsize=(14,6))
        plt.subplot(1,2,1)
        value_1 = df["PerformanceRating"].value_counts()
        plt.title("Employees by PerformanceRating", fontweight="black", size=20, pad=20)
        plt.pie(value_1.values, labels=value_1.index, autopct="%.1f%%",pctdistance=0.75,startangle=90,
            colors=["#b86607","#ff8902","#ffb563","#fad6ac"],textprops={"fontweight":"black","size":15})
        center_circle = plt.Circle((0, 0), 0.4, fc='white')
        fig = plt.gcf()
        fig.gca().add_artist(center_circle)
        #Visualization to show Attrition Rate by PerformanceRating.
        plt.subplot(1,2,2)
        new_df = df[df["Attrition"]=="Yes"]
        value_2 = new_df["PerformanceRating"].value_counts()
        attrition_rate = np.floor((value_2/value_1)*100).values
        sns.barplot(x=value_2.index.tolist(),y= value_2.values,palette=["#D4A1E7","#E7A1A1"])
        plt.title("Attrition Rate by PerformanceRating",fontweight="black",size=20,pad=20)
        for index,value in enumerate(value_2):
            plt.text(index,value,str(value)+" ("+str(int(attrition_rate[index]))+"%)",ha="center",
                    va="bottom",size=15,fontweight="black")
        plt.tight_layout()
        st.pyplot(plt)
    else:
        st.error("The Uploaded Dataset Doesn't contain 'PerformanceRating' columns.")
```

Application > HR.py > ...

```python
# Statistical Analysis - Feature Importance
if 'Statistical Analysis' in visuals:
    all_features = ['Visualizing the F_Score of ANOVA Test of Each Numerical features',
                    'Comparing F_Score and P_value of ANOVA Test',
                    'Visualizing the Chi-Square Statistic Values of Each Categorical Features',
                    'Compairing Chi2_Statistic and P_value of Chi_Square Test']
    features = st.sidebar.multiselect("Choose which features you want to explore on data 👇",all_features)


    st.header("_Statistical Analysis:_")
    if 'Visualizing the F_Score of ANOVA Test of Each Numerical features' in features:

        num_cols = df.select_dtypes(np.number).columns
        new_df = df.copy()
        new_df["Attrition"] = new_df["Attrition"].replace({"No":0,"Yes":1})
        f_scores = {}
        p_values = {}

        for column in num_cols:
            f_score, p_value = stats.f_oneway(new_df[column],new_df["Attrition"])

            f_scores[column] = f_score
            p_values[column] = p_value

        plt.figure(figsize=(15,6))
        keys = list(f_scores.keys())
        values = list(f_scores.values())

        sns.barplot(x=keys, y=values)
        plt.title("  (function) rotation: Literal[90]  ntweight="black",size=20,pad=15)
        plt.xticks(rotation=90)

        for index,value in enumerate(values):
            plt.text(index,value,int(value), ha="center", va="bottom",fontweight="black",size=15)
        #plt.tight_layout()
        st.pyplot(plt)

    if 'Comparing F_Score and P_value of ANOVA Test' in features:
        test_df = pd.DataFrame({"Features":keys,"F_Score":values})
        test_df["P_value"] = [format(p, '.20f') for p in list(p_values.values())]
        st.write(test_df)
```

5.2 User Interface Design





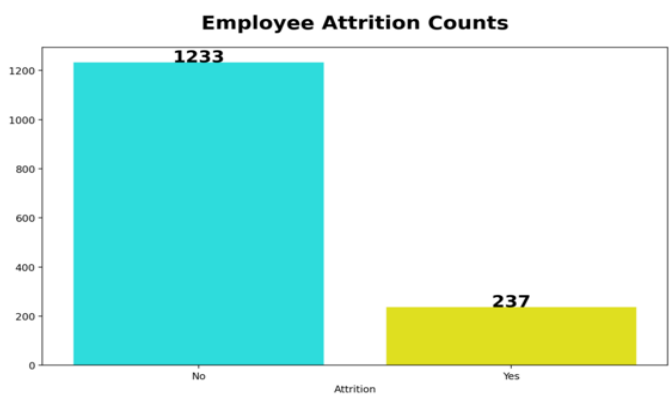
DataSet:

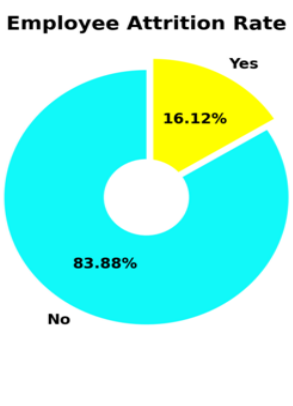
Dataset contains 1470 rows and 32 columns.

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField
0	41	Yes	Travel_Rarely	1,102	Sales	1	2	Life Sciences
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences
2	37	Yes	Travel_Rarely	1,373	Research & Development	2	2	Other
3	33	No	Travel_Frequently	1,392	Research & Development	3	4	Life Sciences
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical
5	32	No	Travel_Frequently	1,005	Research & Development	2	2	Life Sciences
6	59	No	Travel_Rarely	1,324	Research & Development	3	3	Medical
7	30	No	Travel_Rarely	1,358	Research & Development	24	1	Life Sciences
8	38	No	Travel_Frequently	216	Research & Development	23	3	Life Sciences
9	36	No	Travel_Rarely	1,299	Research & Development	27	3	Medical

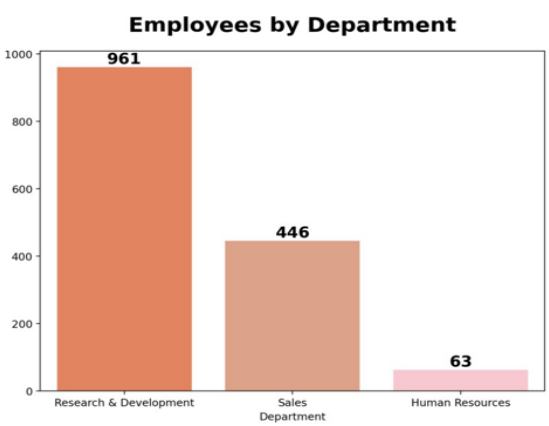
Exploratory Data Analysis:

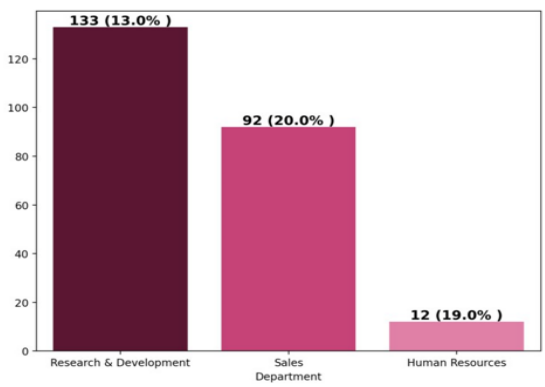
Employee Distribution by Attrition



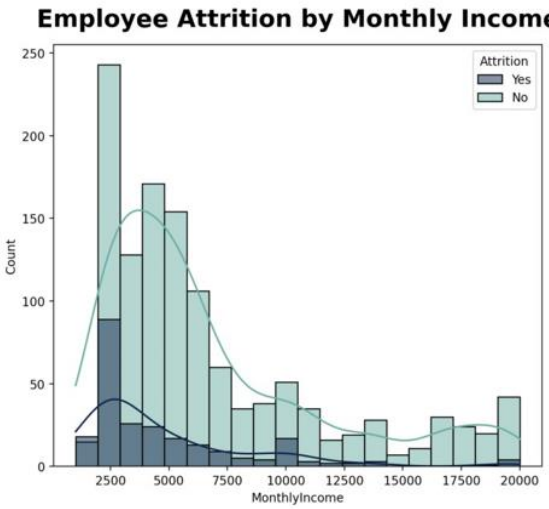


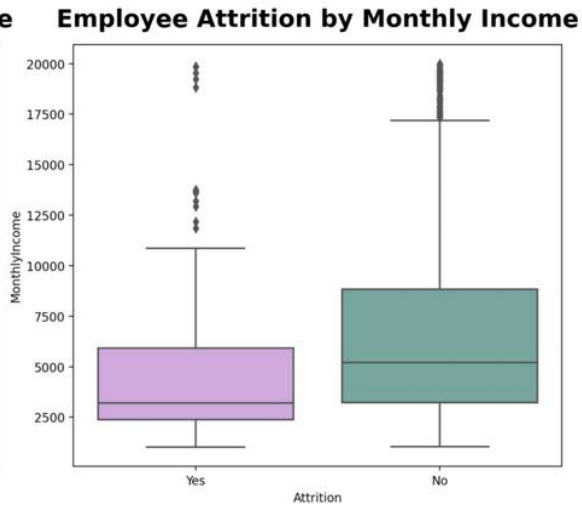
Employees Distribution by Department



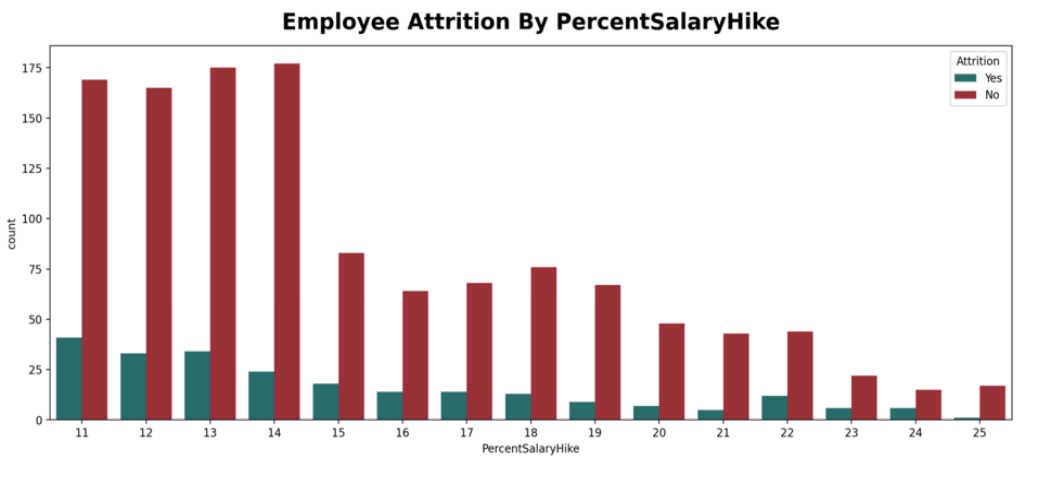


Employee Distribution by Monthly Income

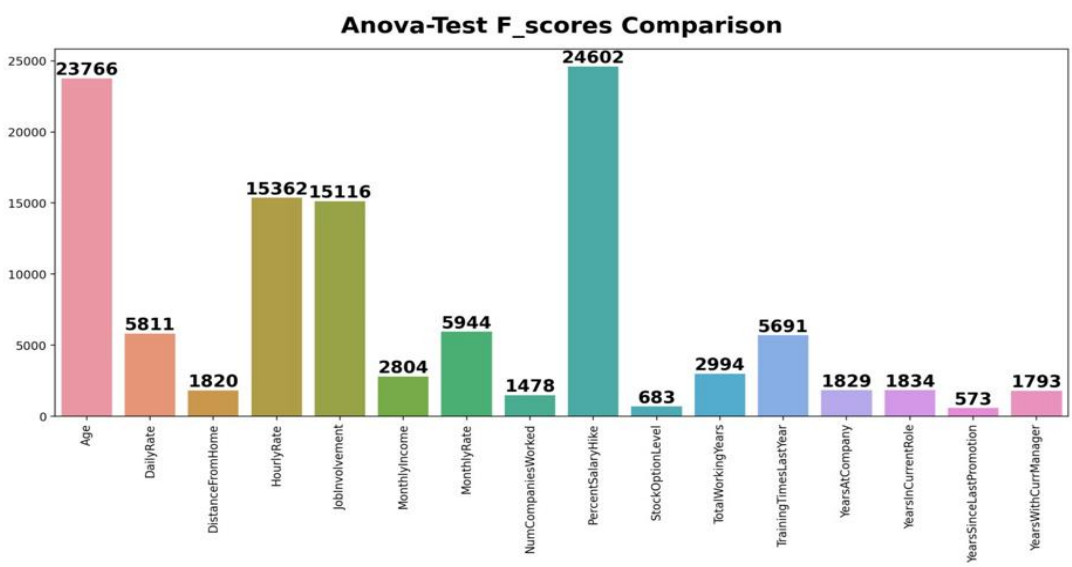




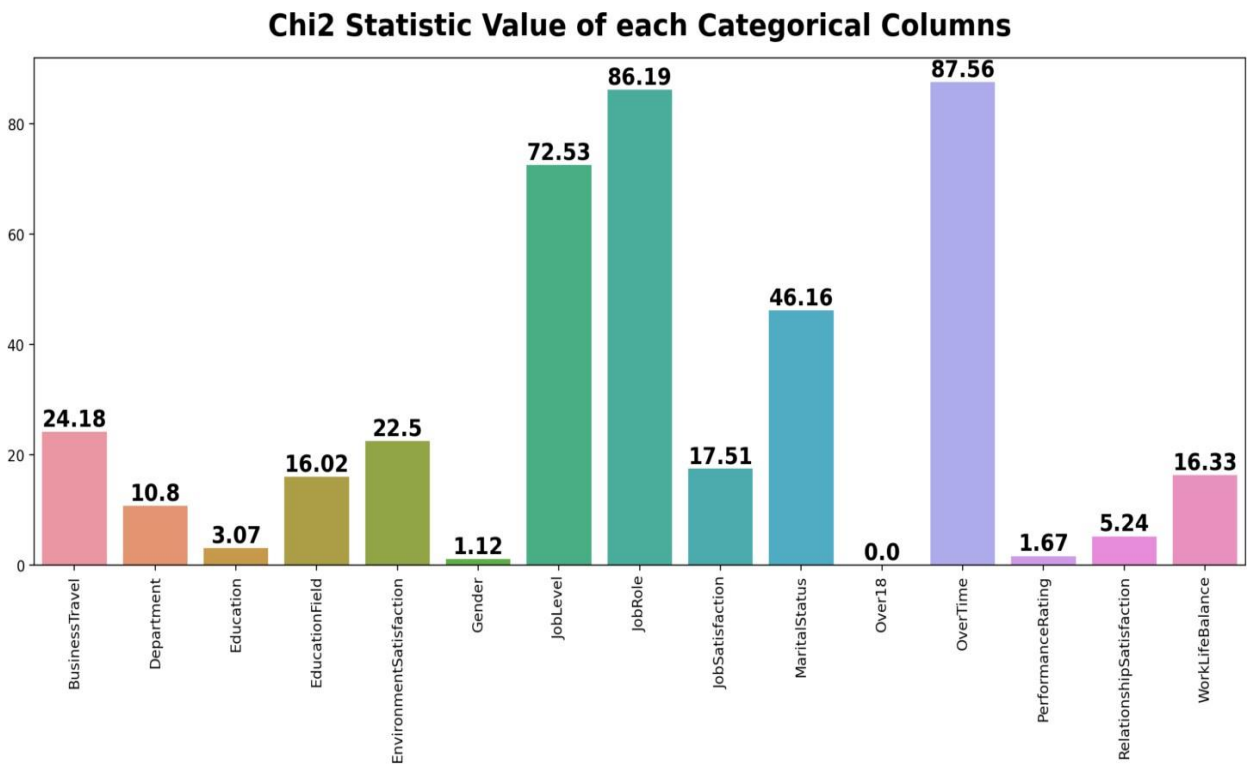
Employee Distribution by Percentage Salary Hike



Statistical Analysis:



	Features	F_Score	P_value
0	Age	23,766.934	0.000000000000000000(
1	DailyRate	5,811.7966	0.000000000000000000(
2	DistanceFromHome	1,820.6146	0.000000000000000000(
3	HourlyRate	15,362.1224	0.000000000000000000(
4	JobInvolvement	15,116.5796	0.000000000000000000(
5	MonthlyIncome	2,804.4596	0.000000000000000000(
6	MonthlyRate	5,944.0891	0.000000000000000000(
7	NumCompaniesWorked	1,478.1886	0.000000000000000000(
8	PercentSalaryHike	24,602.5079	0.000000000000000000(
9	StockOptionLevel	683.0696	0.000000000000000000(



5.3 Test Cases

Test Case ID	*Test Case*	*Test Condition*	*Streps to Executed*	*Expected Result*	*Actual Result*	*Pass /Fail*
TC01	Upload Dataset	User want to upload a dataset to the HR attrition and performance dashboard website	1. Navigate to the HR attrition and performance dashboard website. 2. Locate the option/button to upload a dataset. 3. Click on the upload dataset option. 4. Choose the dataset file from the local system.	- The dataset should be successfully uploaded without any errors. - The uploaded dataset should be visible and accessible for further analysis on the dashboard.	- Dataset should be successfully uploaded without any errors.	Pass
TC02	Select Multiple Points in Sidebar	User want to select multiple points in the sidebar to display them in the dataset and graph.	1. Locate the sidebar containing categories. 2. Select multiple points from the sidebar by clicking. 3. Verify that the selected points are highlighted sidebar.	- Multiple points should be selectable simultaneously without any issues. - Selected points should be visually distinguishable from the unselected ones in the sidebar.	- Multiple point selected simultaneously and also selected point display on sidebar.	Pass
TC03	Display Graphs based on Selected Points	User wants to display graphs based on the points selected in the sidebar	1. Observe the dashboard area where graphs are displayed. 2. Check for the display of graphs corresponding to the selected points.	- Graphs relevant to the selected points should be displayed accurately on the dashboard.	- Graphs relevant to the selected points display successfully	Pass

TC04	Display Error Messag e for Missing Categori -es.	The dataset doesn’t contain categories	1. Upload a dataset without any categories. 2. Try to select points from the sidebar.	- An error message should be displayed, indicating that the dataset does not contain categories.	- Display Error message if dataset does not contain categories.	Fail

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Test Report

This Web Application is useful for HR manager in organization. This application is user friendly and easy to use. We checked things like uploading data, selecting different options, and seeing graphs based on those choices. Everything worked well, showing that the website can handle data accurately and help users understand HR trends effectively.

CHAPTER 7

CONCLUSIONS

7.1 Conclusion

In conclusion, the HR attrition and performance dashboard website provides a comprehensive solution for monitoring employee turnover and performance within an organization. By offering easily accessible data and insights, it empowers HR professionals and decision-makers to make informed choices to enhance employee retention and productivity. With user-friendly interfaces and intuitive features, this platform serves as a valuable tool in managing human resources effectively. Embracing this dashboard can lead to improved workforce management, better employee satisfaction, and ultimately, greater success for the organization as a whole.

7.2 Future Scope

- HR Attrition Dashboard helps to make work easier and save time of user.
- This Web Application is develop using Python Streamlit framework so we can easily implement Data Mining algorithm in our web application in future.

References

1. Ajit, Pankaj(2016). "Prediction of employee turnover in organizations using machine learning algorithms." algorithms 4.5 : C5.

2. Alao, D., Adeyemo, A.B. (2013).Analyzing employee attrition using decision tree algorithms. Comput. Inf. Syst. Dev. Inform. Allied Res. J. 4.

3. Amir Mohammad EsmaieeliSikaroudi, Rouzbehghousi and Ali Esmaieelisikaroudi, 2015 “A Data Mining Approach To Employee Turnover Prediction” (Case Study: Arak Automotive Parts Manufacturing), Journal Of Industrial And Systems Engineering Volume. 8, No. 4.

4. Dilip Singh Sisodia, SomduttaVishwakarma, AbinashPujahari “Evaluation of Machine Learning Models for Employee Churn Prediction”, Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017) IEEE [13] Xplore Compliant - Part Number: CFP17L34-ART, ISBN: 978-1-5386-4031-9.