PROJECT REPORT

1. INTRODUCTION:

1.1 Project Overview:

In an era marked by unprecedented technological advancements and evolving organizational dynamics, the pursuit of optimal workforce management has become a critical imperative for businesses. Our project, cantered on employee performance prediction, emerges as a pioneering solution at the intersection of data science and human resource management. Leveraging machine learning techniques, we seek to develop a predictive model that not only comprehensively assesses an employee's past performance but also forecasts their future contributions to the organization. Through the judicious analysis of diverse data sources, including historical performance records, professional development trajectories, and interpersonal dynamics, our project aims to empower organizations with the foresight needed to make informed decisions about talent management.

1.2 Purpose:

The purpose of this project is multifaceted. Primarily, it endeavours to address the challenges associated with traditional performance evaluation methods, which often rely on subjective assessments and retrospective analyses. By harnessing the power of machine learning algorithms, our goal is to provide organizations with a robust, data-driven tool that not only identifies key performance indicators but also anticipates future trends. This predictive capability enables proactive decision-making in talent retention, career development, and resource allocation. Moreover, our project aspires to contribute to the broader discourse on the convergence of artificial intelligence and human resources, highlighting the transformative potential of data-driven insights in shaping the future of work. Through this initiative, we seek to foster a culture of innovation and efficiency in workforce management.

2. LITERATURE SURVEY:

2.1 Existing Problem:

The existing landscape of employee performance evaluation is fraught with challenges that demand innovative solutions. Traditional methods, often reliant on manual assessments and subjective judgments, are prone to bias and fail to capture the nuanced dynamics of modern workplaces. Inaccurate evaluations hinder effective talent management and can result in suboptimal resource allocation. Moreover, the rapid evolution of work environments, filled by technological advancements and changing expectations, necessitates a shift towards more dynamic and data-driven performance assessment methods. This literature survey critically examines the shortcomings of current approaches and identifies the pressing need for a comprehensive, predictive model that can adapt to the complexities of contemporary workplaces.

2.2 References:

Here are some references that we use for our project machine learning approach for employee performance prediction:

Research Papers:

A Machine Learning Approach for Employee Retention Prediction: https://www.researchgate.net/publication/355078013_A_Machine_Learning_Approach_for_Employee_Retention_Prediction

Prediction of Employee Performance using Naive Bayes International Journal of Advanced Trends in Computer Science and Engineering: https://www.researchgate.net/publication/338363513_Employee_Performance_Prediction_using_Naive_Bayes

Research on Employee Performance Prediction Based on Machine Learning: https://ieeexplore.ieee.org/document/9824477/

Prediction of employee performance using machine learning techniques: https://dl.acm.org/doi/10.1145/3373477.3373696

Employee Turnover Prediction with Machine Learning: A Reliable Approach: https://www.andrew.cmu.edu/user/yuezhao2/papers/18-intellisys-employee.pdf

Datasets:

UCI Machine Learning Repository - HR Analytics: https://archive.ics.uci.edu/about

Kaggle - HR Analytics: https://www.kaggle.com/code/jacksonchou/hr-analytics

OpenML - HR Management: https://www.openml.org/d/4135

Additional Resources:

Machine Learning for HR Analytics: https://www.ibm.com/blog/new-ai-hr-talent-laws-get-the-attention-of-the-c-suite/

Using Machine Learning for Employee Performance Prediction: https://towardsdatascience.com/will-your-employee-leave-a-machine-learning-model-8484c2a6663e

5 Machine Learning Algorithms for Employee Performance Prediction: https://dl.acm.org/doi/10.1145/3373477.3373696

Books:

Predictive HR Analytics: Transforming People Analytics into Business Impact: https://www.amazon.com/People-Analytics/s?k=People+Analytics

The HR Scorecard: Measuring and Improving Human Capital Performance: https://www.amazon.com/HR-Scorecard-Linking-Strategy-Performance/dp/1578511364

Machine Learning for Managers: https://www.amazon.com/AI-Managers-science-learning-projects/dp/183763114X

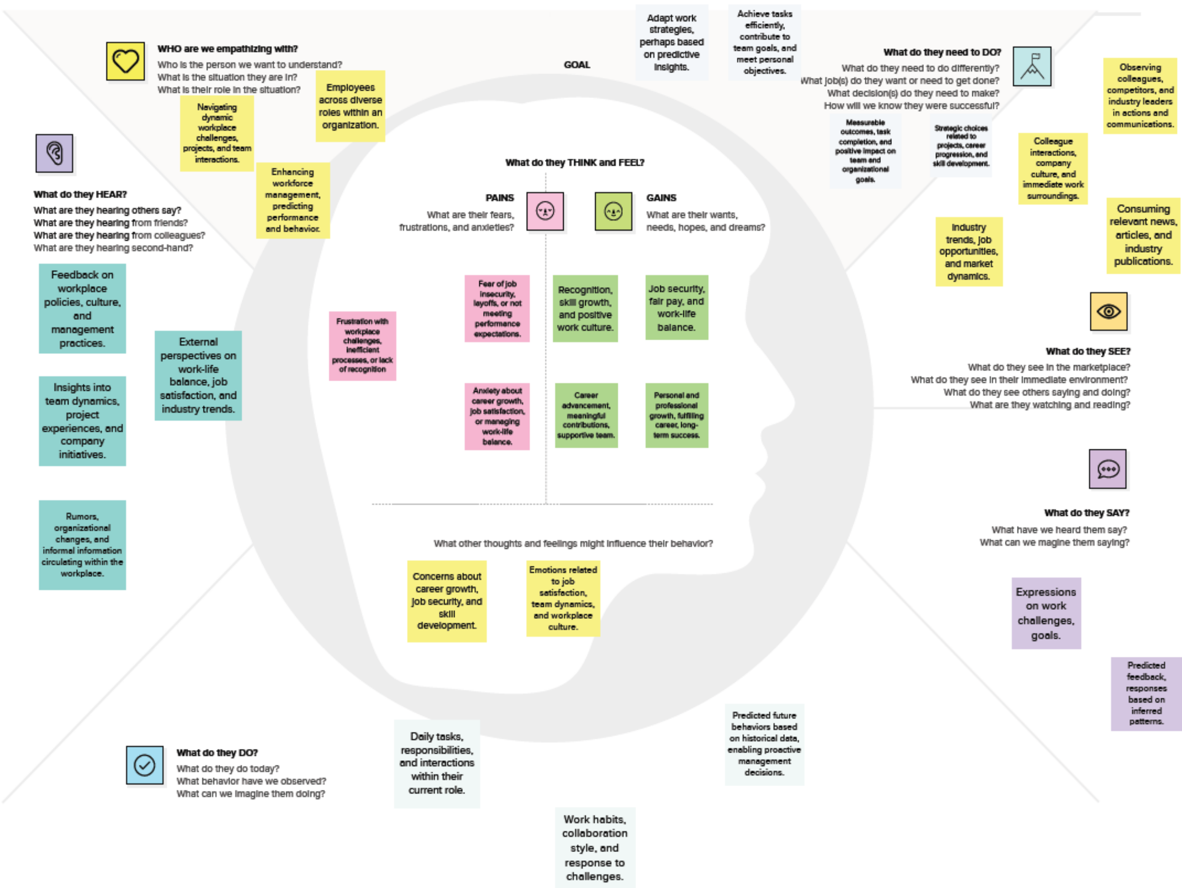
2.3 Problem Statement Definition:

The overarching problem addressed by this project is the inadequacy of existing employee performance evaluation methods in capturing the multifaceted nature of contemporary workplaces. The project aims to develop a machine learning-based model that not only overcomes the limitations of traditional approaches but also anticipates future performance trends. The defined problem statement revolves around the need for a dynamic and predictive performance assessment tool that considers a wide array of factors, including historical performance records, professional development trajectories, and interpersonal dynamics. By tackling this problem, the project seeks to provide organizations with actionable insights for strategic talent management and foster a paradigm shift towards more effective, data-driven HR practices.

3. IDEATION & PROPOSED SOLUTION:

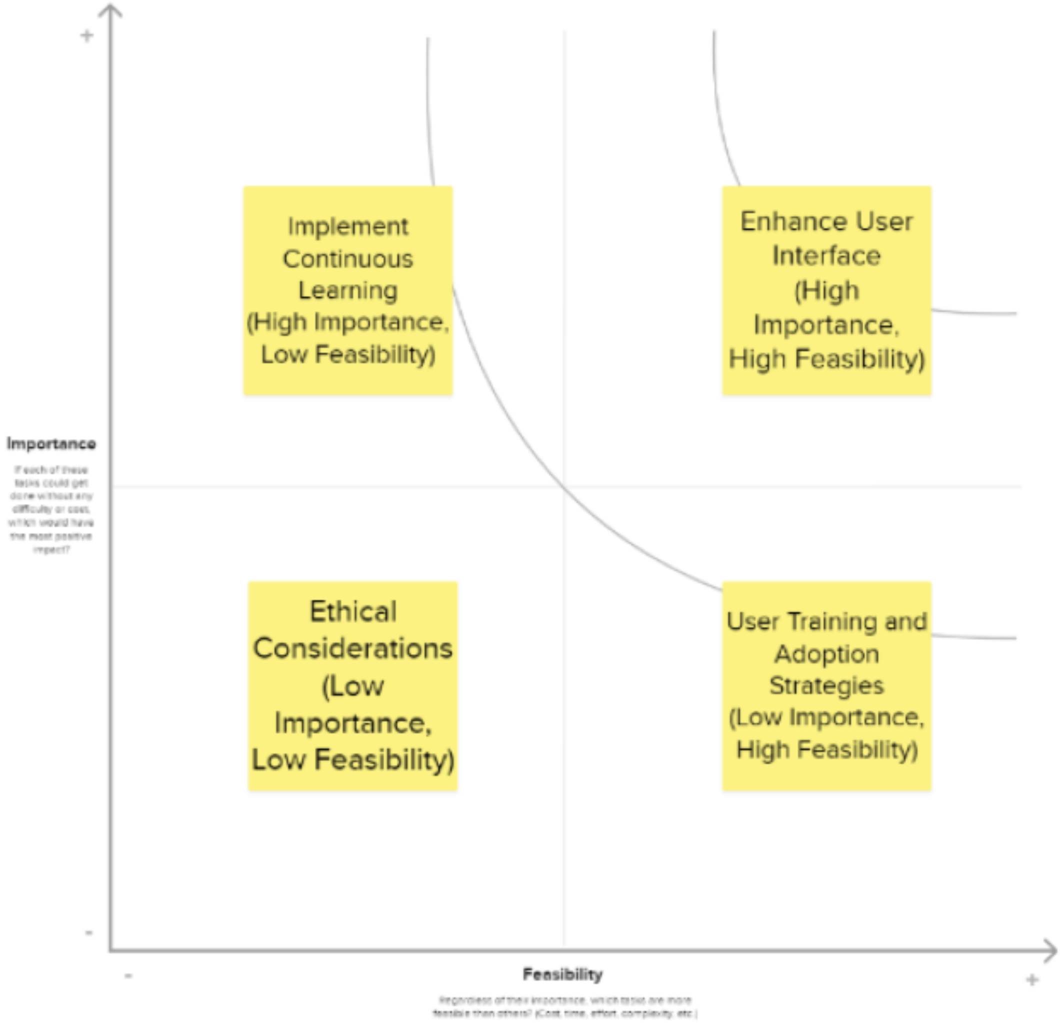
3.1 Empathy Map Canvas:

To comprehend the nuanced needs and expectations of both employers and employees, we employed an Empathy Map Canvas. This tool allowed us to visualize and empathize with the experiences, thoughts, and feelings of all stakeholders involved. Through this exercise, we identified key pain points in the current performance evaluation processes, such as subjective assessments, lack of transparency, and the need for personalized career development.



3.2 Ideation & Brainstorming:

The objective of brainstorming for a Machine Learning Approach for Employee Performance Prediction is to explore and generate innovative ideas across crucial aspects like data integration, algorithm selection, and ethical considerations. By fostering collaboration and diverse perspectives, this process aims to identify potential challenges, innovative solutions, and key elements that will contribute to the development of an effective, transparent, and ethically sound predictive model for optimizing workforce management and decision-making.



In crafting a strategic roadmap for a Machine Learning Approach for Employee Performance Prediction, prioritizing tasks based on both importance and feasibility is paramount. The foremost task, given high importance and feasibility, involves enhancing the user interface. Improving this interface is critical for fostering user understanding and engagement, with a readily achievable implementation requiring moderate effort and time. Next, addressing the high importance but low feasibility task of implementing continuous learning mechanisms demands careful planning and resource allocation. This complex but impactful initiative may require phased execution or innovative solutions to overcome challenges. On the other hand, despite its lower criticality, implementing user training and adoption strategies, characterized by low importance and high feasibility, presents an opportunity for efficient refinement with relatively lower effort and cost. Lastly, the low importance and low feasibility task of addressing ethical considerations, while essential, may be deferred to future phases or advancements in technology due to its intricate nature. This strategic prioritization ensures that efforts are directed towards initiatives that promise both significant positive impact and practical feasibility for optimal success in the implementation of the Machine Learning Approach.

4. REQUIREMENT ANALYSIS

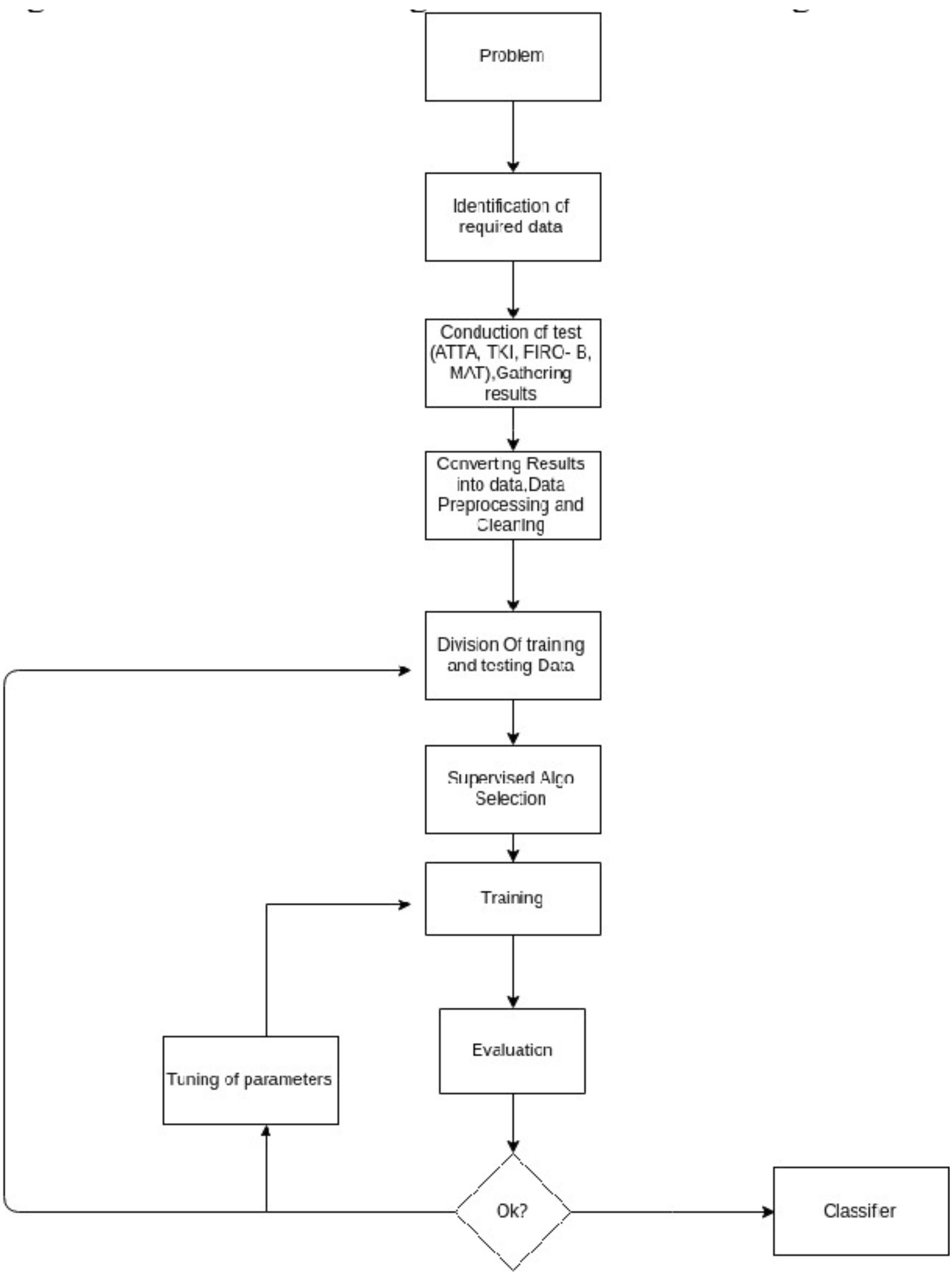
4.1 Functional and Non-Functional Requirements:

The functional requirements of our employee performance prediction system centre on its ability to seamlessly aggregate data from varied sources, including historical performance records and professional development milestones. At its core, the system must deploy a sophisticated machine learning model that comprehensively analyses this amalgamated data, offering accurate predictions of employee performance. The user interface is designed with utmost user-friendliness, providing both employees and managers with an intuitive platform to access and interpret performance metrics. In parallel, the system features robust user authentication, role-based access controls, and a feedback mechanism, ensuring secure and collaborative interaction. On the non-functional front, the system prioritizes performance by guaranteeing swift response times and real-time predictions. Security measures, including data encryption and secure communication, underscore the system's commitment to confidentiality and integrity. Reliability is a cornerstone, minimizing downtime and ensuring continuous availability. Scalability is built into the system's DNA, accommodating growing datasets and user numbers. Additionally, ethical compliance and adherence to data protection regulations are integral to the non-functional requirements, aligning the system with ethical and legal standards.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories:

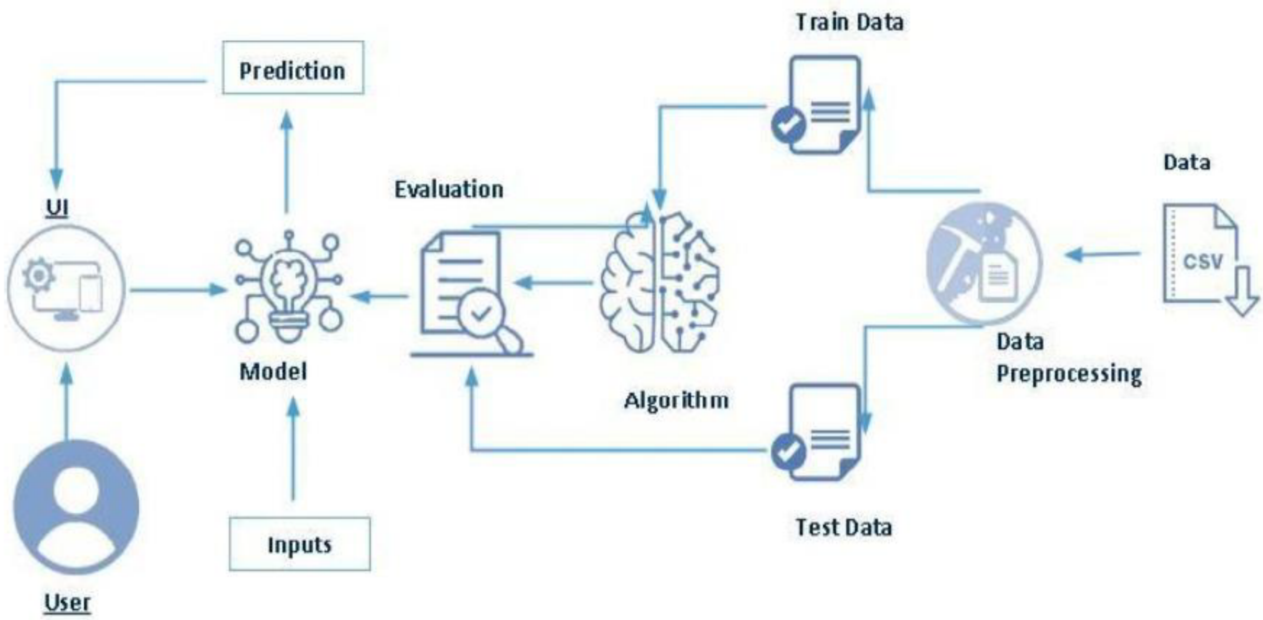
The Data Flow Diagrams (DFDs) and User Stories constitute pivotal components in delineating the functionality and user interactions within our employee performance prediction system. The DFDs visually represent the flow of data within the system, illustrating how information moves between different components and processes. Through User Stories, we encapsulate the system's functionality from the perspective of end-users, detailing their interactions with the system and the value derived. For instance, a User Story may depict a manager accessing the performance dashboard to review individual and team metrics, while a DFD would illustrate the data flow from the database to the visualization component for that specific user action. This synergy between DFDs and User Stories ensures a comprehensive understanding of system behaviour, fostering effective communication between stakeholders and development teams.



5.2 Solution Architecture:

The solution employs a robust architecture, integrating data ingestion, preprocessing, and advanced machine learning models. It emphasizes feature selection, adaptive learning, and ethical considerations. A user-friendly interface facilitates predictions, while scalability and integration APIs ensure seamless adoption. Security measures safeguard sensitive data, contributing to a comprehensive and efficient Machine Learning approach for Employee Performance Prediction.

Solution Architecture Diagram



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture:

The technical architecture for Employee Performance Prediction utilizing machine learning involves a systematic process. Beginning with Data Collection and Preprocessing, APIs are employed to gather information from various sources, such as HR systems, performance reviews, training records, and employee surveys. Data cleansing is performed using Python libraries like Pandas and NumPy to handle missing values, outliers, and inconsistencies, followed by transformation techniques like scaling and normalization to prepare the data for machine learning algorithms. The subsequent step entails Model Development and Training, where appropriate algorithms like Regression (Linear Regression, Lasso Regression) or Classification (Logistic Regression, SVM) are selected for continuous or categorical performance prediction, respectively. Training and validation are conducted using scikit-learn on Python, with hyperparameter tuning for optimal performance. The Deployment and Monitoring phase involves deploying the chosen model into a production environment on cloud platforms like AWS, Azure, or Google Cloud. Integration with existing HR systems via APIs ensures automated predictions, and ongoing monitoring using metrics like accuracy, precision, recall, and F1 score ensures sustained model performance. Visualization and User Interface development employ Python libraries like Matplotlib and Seaborn for dashboards, while user interaction is facilitated through Flask or Dash. The overall architecture facilitates automated performance prediction, data-driven decision-making, improved talent acquisition, and personalized performance management, offering a comprehensive and scalable solution for enhancing organizational efficiency.

```
    +-----------------+
    | Data Sources    |
 +->| (HR systems,   |
 |  | reviews, etc.)|
 +->|                 |
    |  Data Collection & Preprocessing | (Cleaning, normalization, feature engineering)
    |                 |
    +-----------------+
    | Model Development & Training | (Model selection, training, tuning, selection)
    |                 |
    +-----------------+
    | Model Deployment & Monitoring | (API integration, visualization, user interface)
    |                 |
    +-----------------+
    | Predictions    | (Automated performance prediction, data-driven decisions)
    |                 |
    +-----------------+
    | HR Decisions    | (Talent acquisition, performance management, etc.)
```

6.2 Sprint Planning & Estimation:

Sprint planning and estimation are integral to our agile development methodology, ensuring incremental progress and timely delivery. Each sprint, typically spanning two weeks, begins with a collaborative planning session involving cross-functional teams. User Stories, derived from stakeholder requirements, are prioritized and broken down into development tasks. The effort required for each task is estimated through consensus-based techniques such as Planning Poker, fostering team alignment. The sprint backlog, comprising the selected User Stories and tasks, is then committed for completion within the sprint duration. This iterative approach allows for flexibility and adaptability, with regular sprint reviews and retrospectives providing opportunities for continuous improvement. The transparent and collaborative nature of sprint planning and estimation ensures that the project stays on track, meeting stakeholder expectations and adapting to changing priorities.

6.3 Sprint Delivery Schedule:

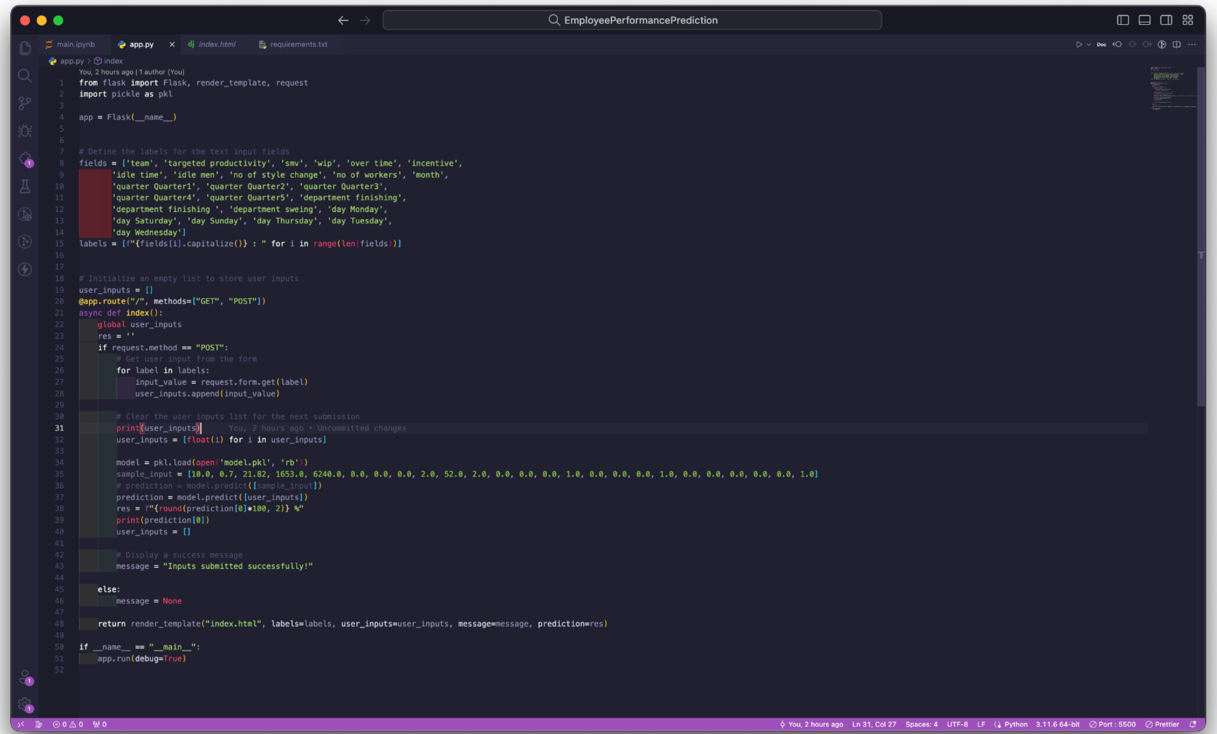
Our sprint delivery schedule follows a consistent and predictable rhythm, with bi-weekly sprint cycles driving the development and delivery process. Each sprint begins with planning and estimation sessions, defining the scope of work to be undertaken. The development phase sees teams collaborating on their assigned tasks, with daily stand-up meetings ensuring real-time communication and issue resolution. Regular testing and quality assurance processes are integrated throughout the sprint, guaranteeing the delivery of high-quality increments. The sprint concludes with a demo showcasing the completed User Stories, providing stakeholders with a tangible view of progress. Subsequent sprint reviews and retrospectives facilitate continuous improvement and refinement of the development process. This structured sprint delivery schedule not only ensures the steady evolution of the employee performance prediction system but also fosters transparency and accountability within the development team.

7. CODING & SOLUTIONING

7.1 Feature 1: Performance Prediction using Regression

Explanation:

The first feature involves implementing Regression using XGBoost. The Flask web application takes user input for key parameters like Day of Week, Quarter of Month, No of Workers, Targeted Productivity, etc(rest are given below in a screenshot). The model, loaded from 'model.pkl', predicts the performance of the employee with the given statistics using the tuned XGBoost Regressor. The result is displayed on the web page.



```python
from flask import Flask, render_template, request
import pickle as pkl

app = Flask(__name__)


# Define the labels for the text input fields
fields = ['team', 'targeted productivity', 'smv', 'wip', 'over time', 'incentive',
        'idle time', 'idle men', 'no of style change', 'no of workers', 'month',
        'quarter Quarter1', 'quarter Quarter2', 'quarter Quarter3',
        'quarter Quarter4', 'quarter Quarter5', 'department finishing',
        'department finishing ', 'department sweing', 'day Monday',
        'day Saturday', 'day Sunday', 'day Thursday', 'day Tuesday',
        'day Wednesday']
labels = [f"{fields[i].capitalize()} : " for i in range(len(fields))]


# Initialize an empty list to store user inputs
user_inputs = []
@app.route("/", methods=["GET", "POST"])
async def index():
    global user_inputs
    res = ''
    if request.method == "POST":
        # Get user input from the form
        for label in labels:
            input_value = request.form.get(label)
            user_inputs.append(input_value)

        # Clear the user inputs list for the next submission
        print(user_inputs)
        user_inputs = [float(i) for i in user_inputs]

        model = pkl.load(open('model.pkl', 'rb'))
        sample_input = [10.0, 0.7, 21.82, 1653.0, 6240.0, 0.0, 0.0, 0.0, 2.0, 52.0, 2.0, 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]
        # prediction = model.predict([sample_input])
        prediction = model.predict([user_inputs])
        res = f"{round(prediction[0]*100, 2)} %"
        print(prediction[0])
        user_inputs = []

        # Display a success message
        message = "Inputs submitted successfully!"

    else:
        message = None

    return render_template("index.html", labels=labels, user_inputs=user_inputs, message=message, prediction=res)

if __name__ == "__main__":
    app.run(debug=True)
```

day : Day of the Week

quarter : A portion of the month. A month was divided into four quarters

department : Associated department with the instance

team_no : Associated team number with the instance

no_of_workers : Number of workers in each team

no_of_style_change : Number of changes in the style of a particular product

targeted_productivity : Targeted productivity set by the Authority for each team for each day.

smv : Standard Minute Value, it is the allocated time for a task

wip : Work in progress. Includes the number of unfinished items for products

over_time : Represents the amount of overtime by each team in minutes

incentive : Represents the amount of financial incentive (in BDT) that enables or motivates a particular course of action.

idle_time : The amount of time when the production was interrupted due to several reasons

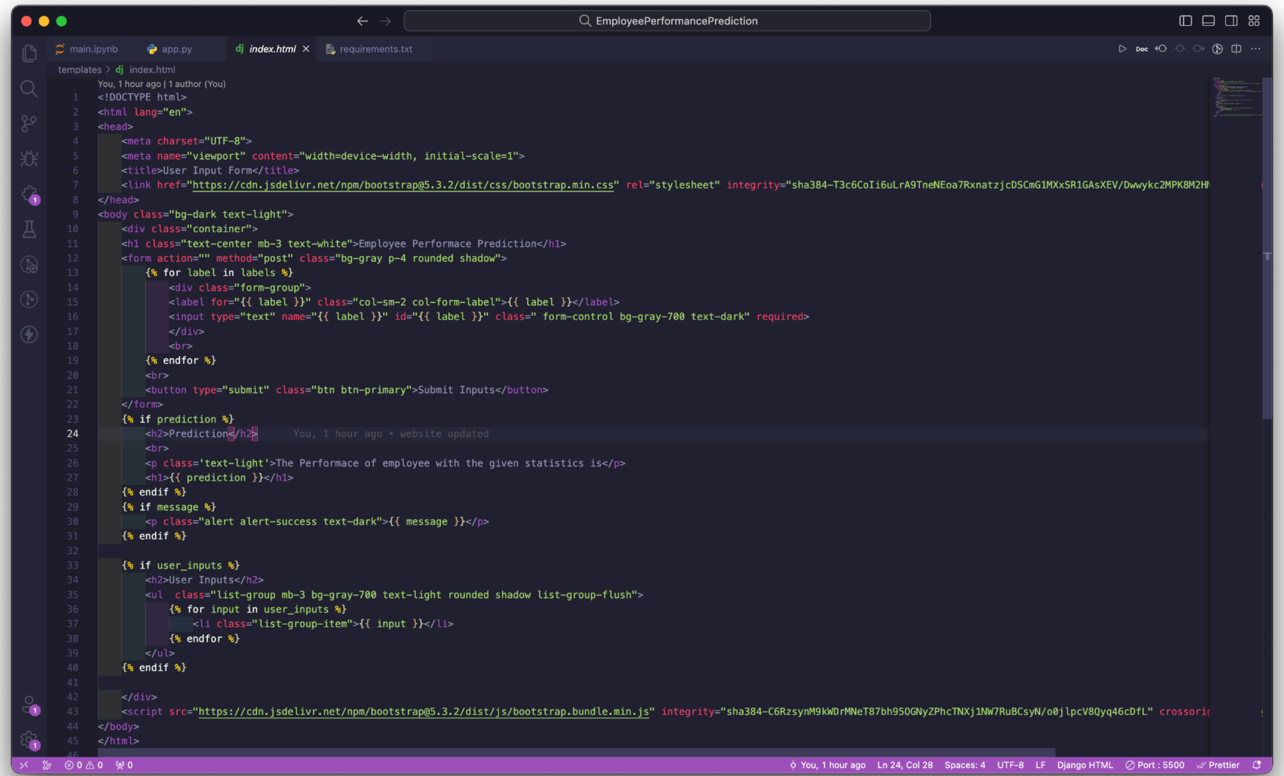
idle_men : The number of workers who were idle due to production interruption

actual_productivity : The actual % of productivity that was delivered by the workers. It ranges from 0-1.

7.2 Feature 2: User-Friendly Interface

Explanation:

Feature 2 focuses on creating a user-friendly interface with HTML styling. The provided HTML code includes styling rules to enhance the visual appeal of the web application using bootstrap. The form elements are designed for ease of use, and the result is displayed in a visually distinct section at the bottom . This feature contributes to a positive user experience when interacting with the application.:

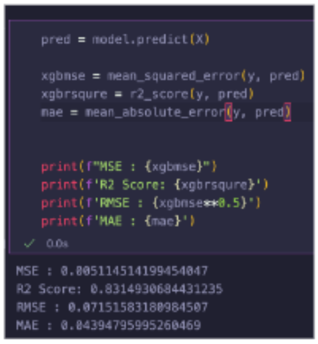
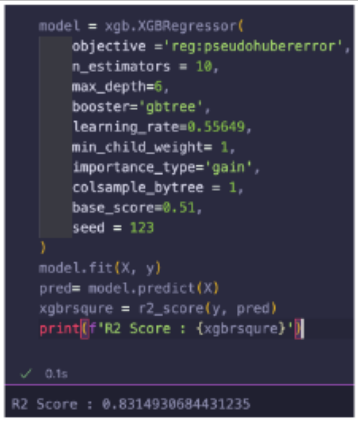


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>User Input Form</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
</head>
<body class="bg-dark text-light">
    <div class="container">
    <h1 class="text-center mb-3 text-white">Employee Performace Prediction</h1>
    <form action="" method="post" class="bg-gray p-4 rounded shadow">
        {% for label in labels %}
            <div class="form-group">
            <label for="{{ label }}" class="col-sm-2 col-form-label">{{ label }}</label>
            <input type="text" name="{{ label }}" id="{{ label }}" class=" form-control bg-gray-700 text-dark" required>
            </div>
            <br>
        {% endfor %}
        <br>
        <button type="submit" class="btn btn-primary">Submit Inputs</button>
    </form>
    {% if prediction %}
        <h2>Prediction</h2>
        <br>
        <p class='text-light'>The Performace of employee with the given statistics is</p>
        <h1>{{ prediction }}</h1>
    {% endif %}
    {% if message %}
        <p class="alert alert-success text-dark">{{ message }}</p>
    {% endif %}

    {% if user_inputs %}
        <h2>User Inputs</h2>
        <ul  class="list-group mb-3 bg-gray-700 text-light rounded shadow list-group-flush">
            {% for input in user_inputs %}
                <li class="list-group-item">{{ input }}</li>
            {% endfor %}
        </ul>
    {% endif %}

    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorig
</body>
</html>
```

8. Performance Testing:

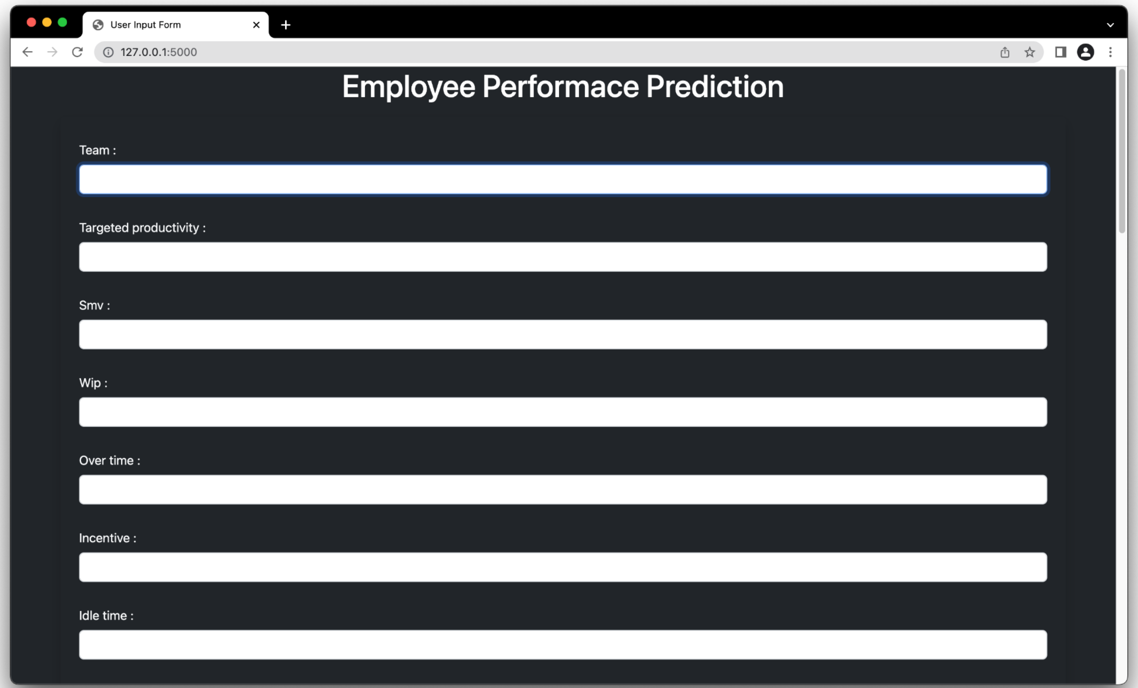
8.1 Performance Matrices:

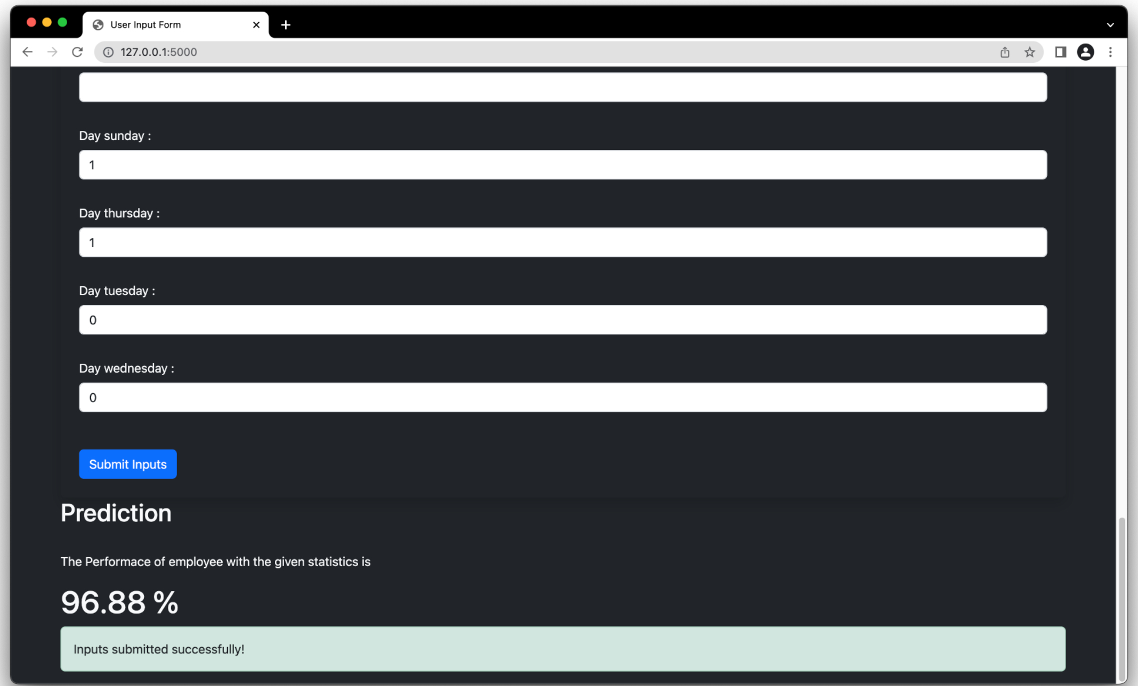
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MSE : 0.005114514199454047 R2 Score: 0.8314930684431235 RMSE : 0.07151583180984507 MAE : 0.04394795995260469	 pred = model.predict(X) xgbmse = mean_squared_error(y, pred) xgbrsqure = r2_score(y, pred) mae = mean_absolute_error(y, pred) print(f"MSE : {xgbmse}") print(f'R2 Score: {xgbrsqure}') print(f'RMSE : {xgbmse**0.5}') print(f'MAE : {mae}') 0.0s MSE : 0.005114514199454047 R2 Score: 0.8314930684431235 RMSE : 0.07151583180984507 MAE : 0.04394795995260469
2.	Tune the Model	Hyperparameter Tuning - objective ='reg:pseudohubererror', n_estimators = 10, max_depth=6, booster='gbtree', learning_rate=0.55649, min_child_weight= 1, importance_type='gain', colsample_bytree = 1, base_score=0.51 Validation Method - train_test_split	 model = xgb.XGBRegressor(objective ='reg:pseudohubererror', n_estimators = 10, max_depth=6, booster='gbtree', learning_rate=0.55649, min_child_weight= 1, importance_type='gain', colsample_bytree = 1, base_score=0.51, seed = 123) model.fit(X, y) pred= model.predict(X) xgbrsqure = r2_score(y, pred) print(f'R2 Score : {xgbrsqure}') 0.1s R2 Score : 0.8314930684431235

9. Results:

9.1 Output Screenshots:





10. Advantages & Disadvantages:

Advantages:

The technical architecture designed for Employee Performance Prediction brings several notable advantages to organizational processes. Firstly, it introduces automation into decision-making, streamlining talent management and resource allocation. This automation is further complemented by the ability to derive data-driven insights from diverse sources, facilitating more informed and strategic decision-making. The system's predictive capabilities contribute to improved talent acquisition, assisting in identifying key performance indicators and forecasting future potential. Moreover, the architecture allows for personalized performance management, adapting predictions to the individual characteristics of each employee. Lastly, the implementation enhances operational efficiency by reducing the time and resources traditionally invested in manual performance evaluation processes.

Disadvantages:

However, the adoption of such a system is not without its challenges. Privacy concerns may arise due to the collection and integration of data from various sources, requiring stringent measures to ensure compliance with regulations and ethical standards. Algorithmic bias is another potential drawback, as the machine learning models may inadvertently perpetuate biases present in historical data, necessitating ongoing monitoring and mitigation efforts. The complexity of implementation is an inherent challenge, demanding specialized expertise in data science and machine learning. Additionally, the accuracy and effectiveness of the system heavily rely on the quality of input data, making data quality a critical factor. Finally, resistance to change from individuals accustomed to traditional evaluation methods may pose a hurdle, emphasizing the importance of effective change management strategies.

11. Conclusion:

In conclusion, the technical architecture devised for Employee Performance Prediction through machine learning presents a forward-looking approach to talent management and organizational efficiency. The advantages of automated decision-making, data-driven insights, improved talent acquisition, and personalized performance management underscore its potential transformative impact. However, the system is not without its challenges, including privacy concerns, algorithmic bias, implementation complexity, data quality dependencies, and resistance to change. Mitigating these challenges through rigorous adherence to ethical standards, continuous monitoring, and effective change management strategies is imperative. The architecture's ability to streamline processes, offer predictive insights, and enhance efficiency is poised to reshape how organizations approach employee performance evaluation. As technological advancements continue, the successful integration and evolution of this architecture will be pivotal in fostering a culture of innovation, fairness, and informed decision-making within the realm of human resource management.

12. Future Scope:

The future scope of the Employee Performance Prediction system is exceptionally promising, heralding a new era in human resource management innovation. The integration of advanced machine learning techniques, particularly delving into the realms of deep learning algorithms, holds the potential to significantly elevate prediction accuracy and unravel intricate relationships within diverse data sources. To further enhance transparency and user trust, future developments could focus on incorporating features that provide detailed explanations for model predictions, effectively addressing concerns related to algorithmic bias and ensuring fairness. The integration of emerging technologies, such as natural language processing and sentiment analysis, offers an exciting frontier for the system to glean insights from unstructured data like employee feedback, enriching the depth of performance analysis. Continuous monitoring and automatic model updates will be instrumental in maintaining the system's relevance, adapting to evolving organizational dynamics, and ensuring sustained accuracy. Moreover, envisioning features that promote employee engagement, such as personalized development plans based on performance predictions, not only enhances the system's functionality but also reinforces a positive and collaborative organizational culture. In essence, the future evolution of the Employee Performance Prediction system holds the promise of not just transforming but revolutionizing how organizations understand, manage, and leverage their most valuable asset—their workforce.

13. Appendix:

Source Code - Uploaded in git under **Phase-4**

GitHub Link
GitHub - https://github.com/smartinternz02/SI-GuidedProject-615815-1699551568

Project Demo Uploded on smartintrnz