Automated Man Power Allocation By Performance Analysis and Project Categorization For Construction Projects

24-25J-018

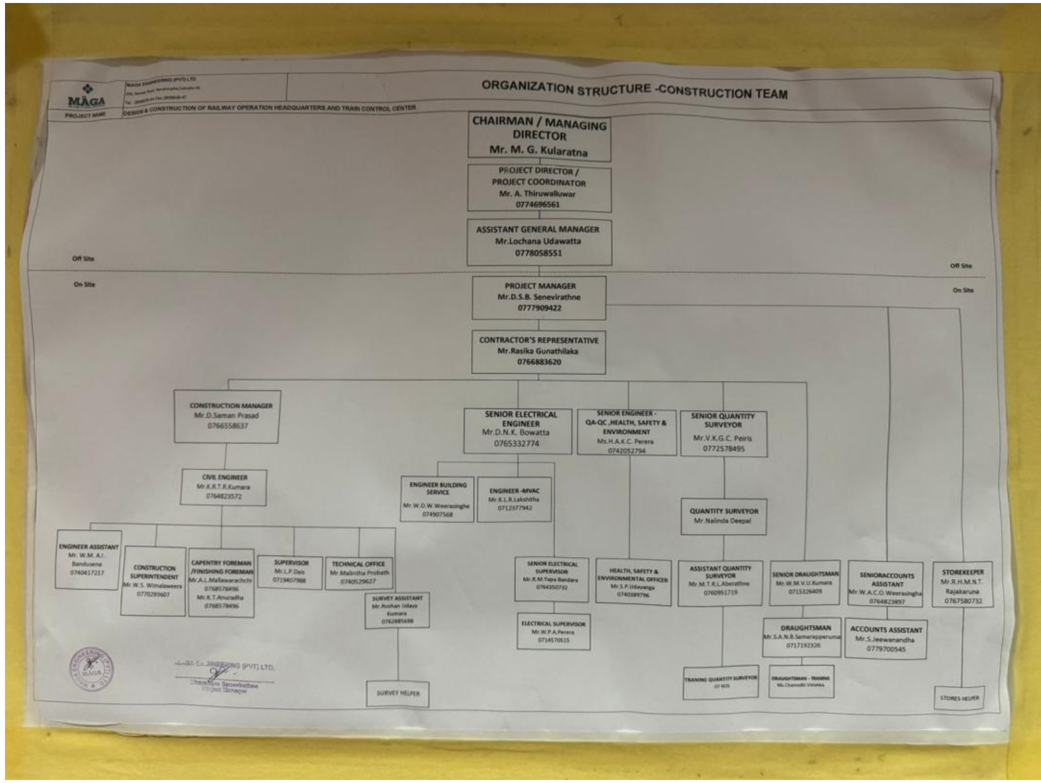
Team Members

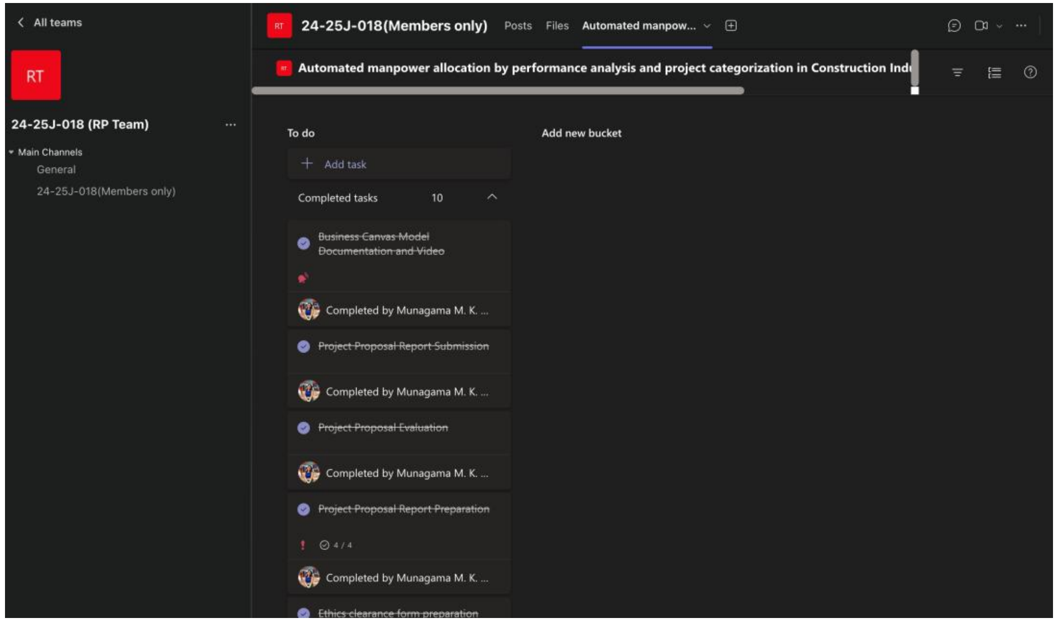
IT Number	Name
IT21270956	Munagama M.K.H
IT21276750	Isuranga K.M.S
IT21069840	Devashika R.P.P.A
IT21087660	Perera K.M

Supervisors

	Name
Supervisor	Ms.Buddima Attanayake
CO-Supervisor	Ms.Narmada Gamage
External Supervisor	Mr.Darshana Senevirathne

Evidences





SLIIT
FACULTY OF COMPUTING

Introduction

- Manpower required in construction projects are in 2 types;

  1. Employees in the Company

  Ex: Civil Engineers,Technical Officers,Surveyors etc.

  2. Laborers work in the site

  Ex:Carpentors,Masons,Painters etc.

- Our client : MAGA Engineering PVT LTD
- Types of construction projects in MAGA;
  1. **Buildings**
  2. Highways & Bridges
  3. Water, wastewater
  4. Irrigation

MÄGA
the saga of quality construction

Research Problem

"How does improper manpower allocation based on project managers' experience affect efficiency and project outcomes in the construction projects?"

Main Objective

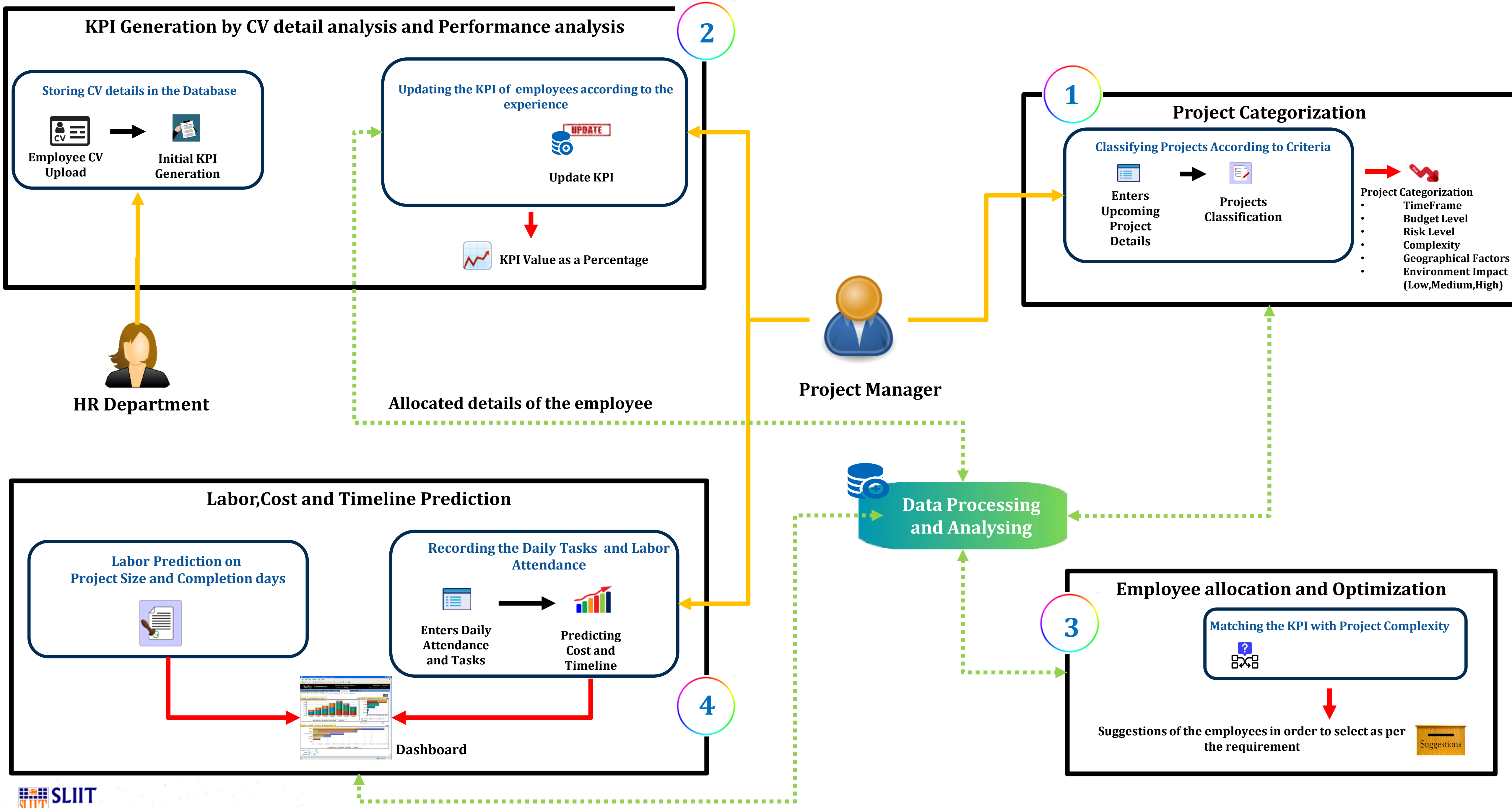
"Develop a system to generate employee KPIs based on experience and performance, categorize projects by complexity and risk, optimally allocate suitable employees, and predict the required number of laborers for future projects."

Sub Objectives

1. Project Categorization.
2. KPI Value generation by performance analysis and CV analysis.
3. Employee allocation and Optimization.
4. Prediction of labor, cost and timeline.

Research Gap

Application Reference	Applicable for construction Projects	Web Application	KPI based manpower allocation	Project Categorization	Real time predictive analysis of project	Labor requirement Prediction
Procore	✓	✓	✗	✗	✗	✗
Primavera P6	✓	✗	✗	✗	✗	✗
nPlan	✓	✓	✗	✓	✓	✗
BuildTrend	✓	✓	✗	✗	✗	✗
Project Pulse	✓	✓	✓	✓	✓	✓


KPI Generation by CV detail analysis and Performance analysis
2
Storing CV details in the Database
Employee CV Upload
Initial KPI Generation
Updating the KPI of employees according to the experience
UPDATE
Update KPI
KPI Value as a Percentage
HR Department
Allocated details of the employee
Project Manager
1
Project Categorization
Classifying Projects According to Criteria
Enters Upcoming Project Details
Projects Classification
Project Categorization
• TimeFrame
• Budget Level
• Risk Level
• Complexity
• Geographical Factors
• Environment Impact
(Low,Medium,High)
Data Processing and Analysing
Labor,Cost and Timeline Prediction
Labor Prediction on Project Size and Completion days
Recording the Daily Tasks and Labor Attendance
Enters Daily Attendance and Tasks
Predicting Cost and Timeline
4
Dashboard
Employee allocation and Optimization
3
Matching the KPI with Project Complexity
Suggestions of the employees in order to select as per the requirement
Suggestions

Project Categorization.

Project Manager

Overview Diagram

Classifying Projects According to Criteria

Enters Upcoming Project Details

- Project Type
- Scope
- Objectives
- Location
- Budget
- Duration
- Site Conditions

Projects Classification

Project Categorization

- Risk Level
- Complexity (Low,Medium,High)

Progress 100% Completion

- Developed a system to enter upcoming project details.
- Classified projects according to specific criteria.
- Created a standardized method for categorizing projects based on risk and complexity using details like type, scope, objectives, location, budget, timeline, and site conditions.
- Implemented project classification based on risk level, complexity, geographical factors, and environmental impact.
- Used Decision Tree Regressor and other models to analyze project risk and complexity.
- Data gathered from MAGA Engineering Pvt Ltd.
- Application reference to tools like Procore, Primavera P6, nPlan, Microsoft Project.

Importance & Industry Benefits:

- Enables automatic classification of construction projects, improving project planning and resource allocation.
- Helps assess environmental impact and geographical considerations effectively.
- Facilitates risk level and complexity assessment that supports better project management.
- Supports integration with existing project management tools for real-time decision-making.
- Reduces manual errors and reliance on subjective project manager experience.

Project Evidence

```
[46] import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import pickle
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.metrics import accuracy_score, classification_report
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler, LabelEncoder
     from sklearn.metrics import classification_report
     from sklearn.tree import DecisionTreeClassifier
```

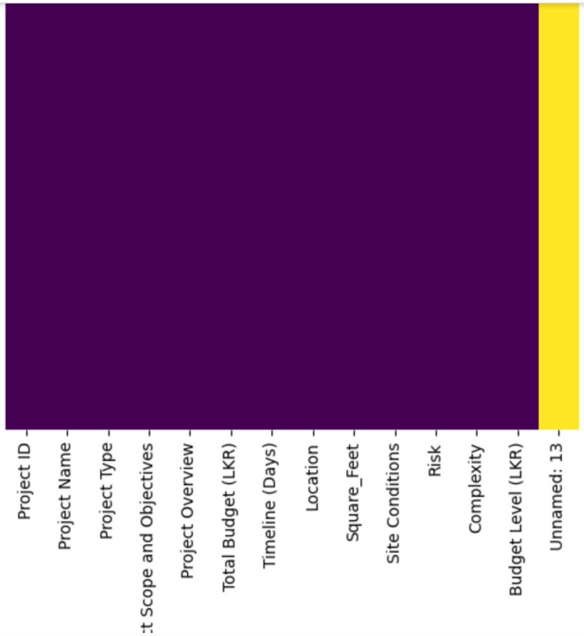
```
[47] data = pd.read_csv('Project Details.csv', encoding='latin-1')
```

```
[48] data
```

	Project ID	Project Name	Project Type	Project Scope and Objectives	Project Overview	Total Budget (LKR)	Timeline (Days)	Location	Square_Feet	Site Conditions	Risk	Complexity	Budg Lev (LK
0	PID - NO -CP - 1	Project 1	Commercial	Develop an eco-friendly resort in the central ...	Construction of 50 villas with solar power and...	Rs. 60,000,000	540	Kandy	15607	Hilly area	Low	High	L

A three

✓ 0s completed at 07:06



+ Code + Text

[51]	
Complexity	0
Budget Level (LKR)	0

dtype: int64

```
[52] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 13 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Project ID                  100 non-null    object
 1   Project Name                100 non-null    object
 2   Project Type                100 non-null    object
 3   Project Scope and Objectives 100 non-null   object
 4   Project Overview            100 non-null    object
 5   Total Budget (LKR)          100 non-null    object
 6   Timeline (Days)             100 non-null    int64
 7   Location                    100 non-null    object
 8   Square_Feet                 100 non-null    int64
 9   Site Conditions             100 non-null    object
 10  Risk                        100 non-null    object
 11  Complexity                  100 non-null    object
 12  Budget Level (LKR)          100 non-null    object
dtypes: int64(2), object(11)
memory usage: 10.3+ KB
```

Total Budg... 76
Timeline (D... 6767
Location: Urban
Square Feet: 2737
Site Conditi... Sloped
Submit

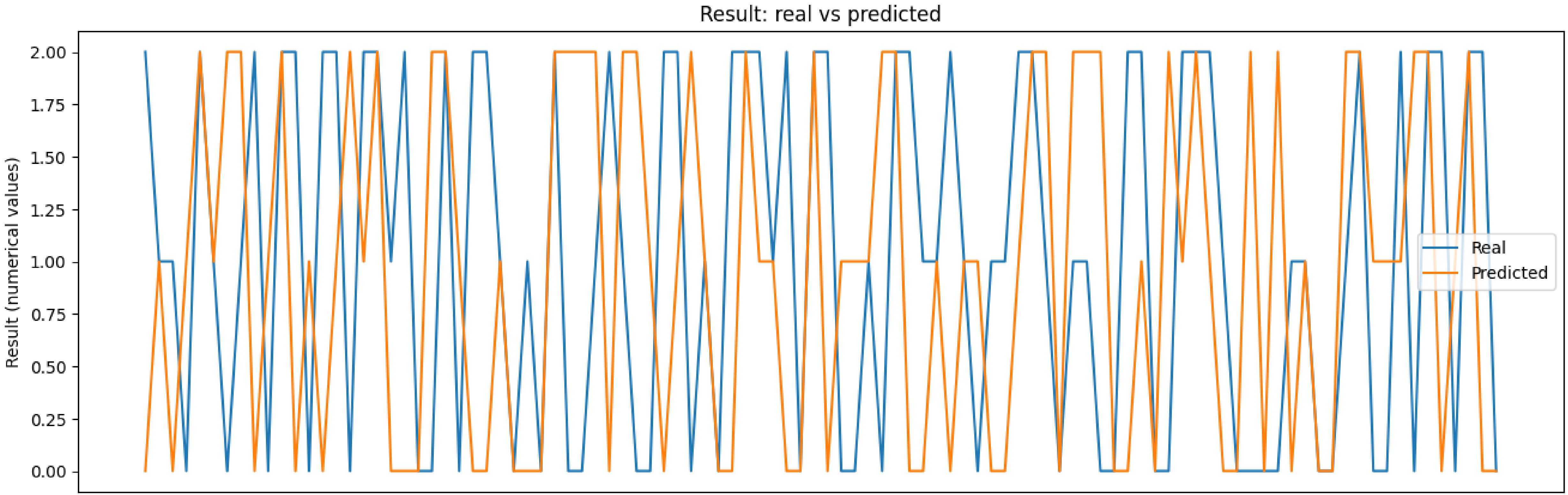
```
Predicted Risk: Medium
Predicted Complexity: Medium
```

```
    total_budget = budget_widget.value
    timeline_days = timeline_widget.value
    square_feet = square_feet_widget.value

    input_data = np.array([[total_budget, timeline_days, location_num, square_feet, site_conditions_num]])

    prediction = model.predict(input_data)

    print("Predicted Risk:", prediction[0][0])
    print("Predicted Complexity:", prediction[0][1])

submit_button = widgets.Button(description="Submit")
submit_button.on_click(submit_input)


display(budget_widget, timeline_widget, location_widget, square_feet_widget, site_conditions_widget, submit_button)
```

Total Budg... 0
Timeline (D... 0
Geographi... Urban
Square Feet: 0
Site Conditi... Flat terrain
Submit

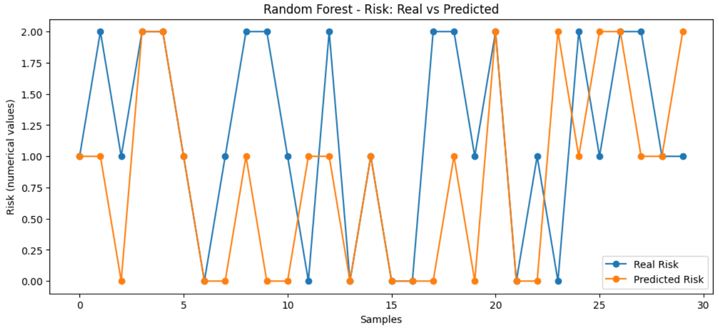
IT21067660 | Perera K.M | 24-25J-0

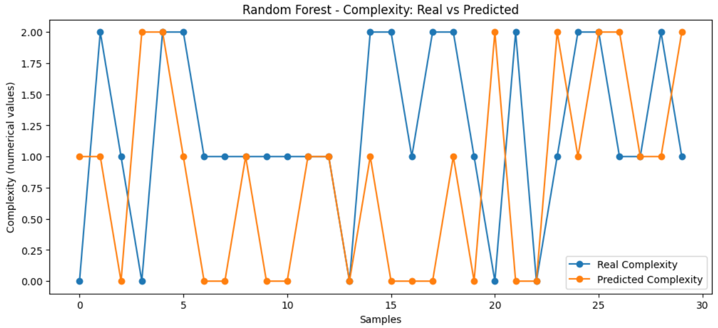
Project Evidence



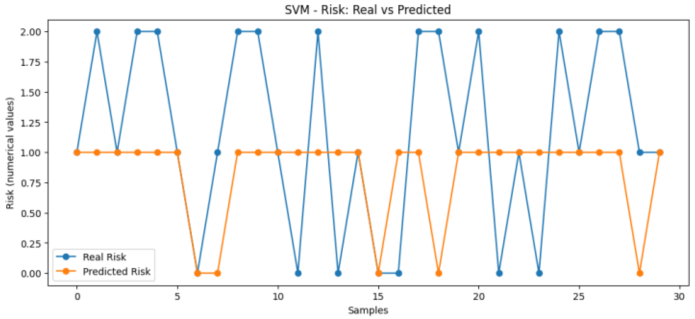
Decision Tree (Both Risk and Complexity)

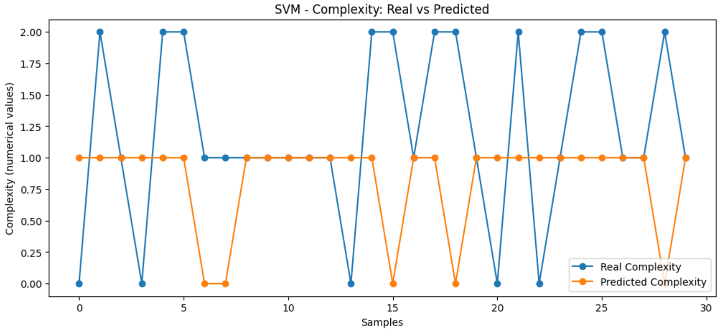
Project Evidence



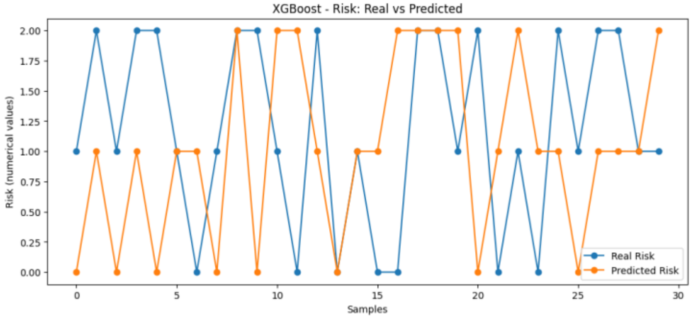


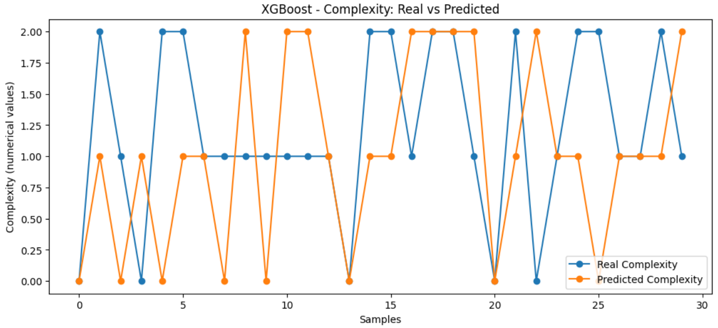
Random Forest





SVM





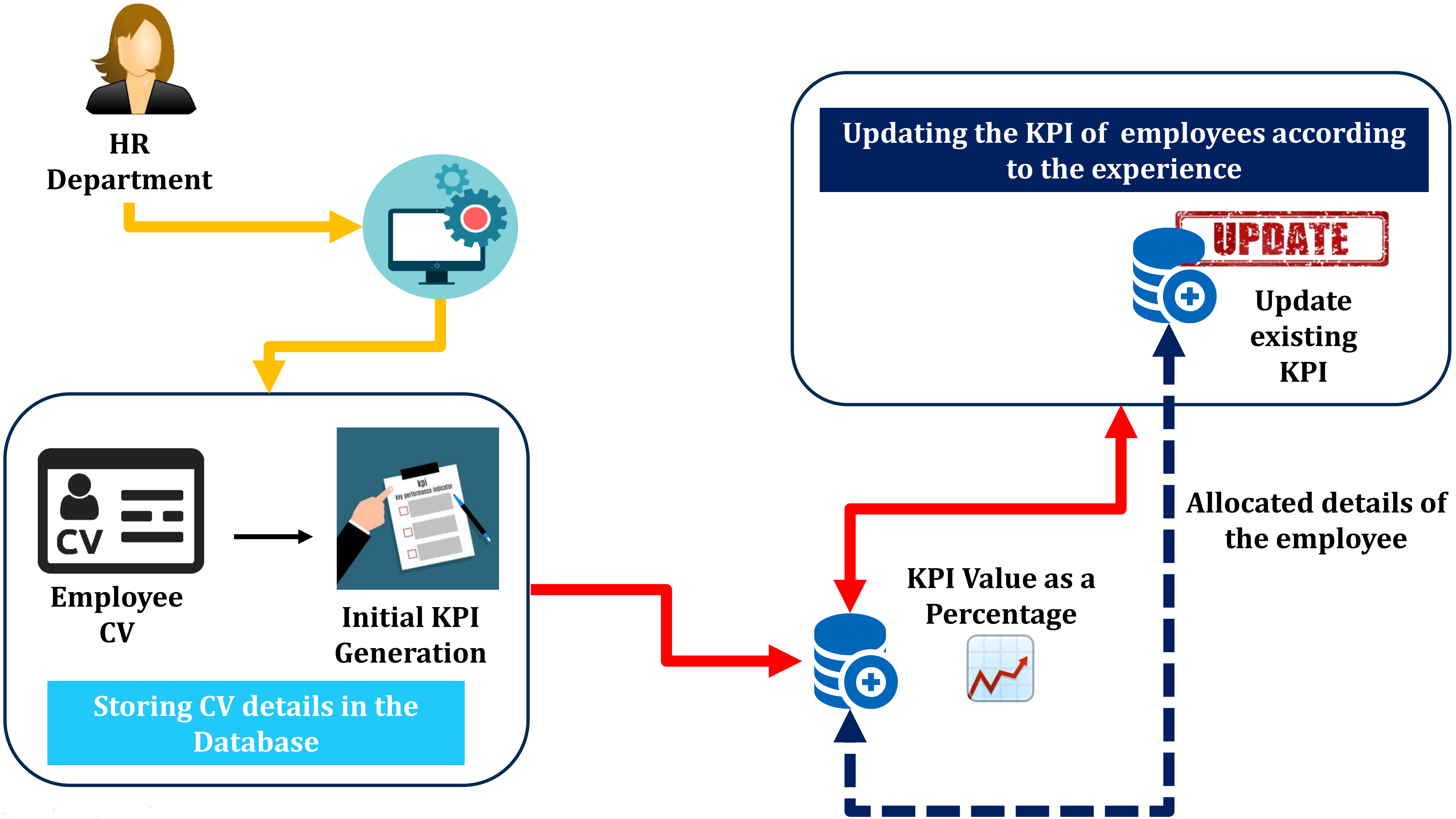
XGBoost

IT21276750|ISURANGA K.M.S

BSc(Hons)in Information Technology specialized in Information Systems Engineering

KPI Generation by Performance analysis and CV analysis.

Overview Diagram


HR
Department
Updating the KPI of employees according to the experience
UPDATE
Update existing KPI
Allocated details of the employee
CV
Employee CV
kpi
key performance indicator
Initial KPI Generation
Storing CV details in the Database
KPI Value as a Percentage

Progress (100% Completion):

- Developed an upload portal for employee CVs.
- Generated initial KPI values from CV data.
- Stored CV details and KPI values in a database.
- Developed functionality to update KPIs based on employee experience and performance.
- Calculated KPI values as percentages for better comparison.
- Applied Natural Language Processing (NLP) and Multi-Nominal Naive Bayes algorithm for analysis.
- Trained models including Random Forest and Decision Tree Regression with decent accuracy.
- Data sourced from MAGA Pvt Ltd.
- References to industry tools like SAP Success Factors, Workday, BambooHR.

Importance & Industry Benefits:

- Automates employee KPI generation, improving accuracy and timeliness.
- Facilitates real-time KPI updates for better HR and project management decisions.
- Reduces challenges related to physical CV management and outdated KPI systems.
- Supports objective employee performance measurement and better manpower planning.
- Integration with HR systems enhances operational efficiency and talent management.

+ Code + Text

```
[2] import json
    import os

    import pandas as  pd
    import spacy

    import seaborn as sns
    import string

    from tqdm import tqdm
    from textblob import TextBlob

    from nltk.corpus import stopwords
    import nltk
    from nltk.stem import WordNetLemmatizer
    from nltk import word_tokenize
    import re

    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder
    from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.feature_extraction.text import TfidfTransformer
    from sklearn.naive_bayes import MultinomialNB
    from sklearn.pipeline import Pipeline


    from sklearn.preprocessing import FunctionTransformer
    from sklearn.base import BaseEstimator, TransformerMixin
    from sklearn.pipeline import FeatureUnion
```

Project Evidence (PP1)

```
[3] data = pd.read_csv('resume.csv')
```

```
[4] data.head()
```

	Resume Summary	KPI Value
0	Name: Jane Smith, Job Title: Construction Fore...	32
1	Name: Daniel Johnson, Job Title: Site Supervis...	39
2	Name: Emily Johnson, Job Title: Civil Engineer...	24
3	Name: Michael Miller, Job Title: Project Manag...	29
4	Name: Jane Garcia, Job Title: Architect, Years...	25

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
[5] data.shape
```

```
(1000, 2)
```

```
[6] data['KPI Value'].value_counts()
```

```
        count
KPI Value
```

```
[7] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Resume Summary  1000 non-null   object
 1   KPI Value       1000 non-null   int64
dtypes: int64(1), object(1)
memory usage: 15.8+ KB
```

```
[8] data['Resume Summary'] = data['Resume Summary'].astype(str)
```

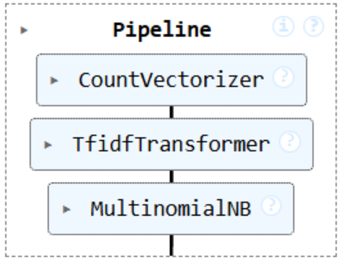
```
[9] import re
    def cleanResume(txt):
        cleanText = re.sub('http\S+\s', ' ', txt)
        cleanText = re.sub('RT|cc', ' ', cleanText)
        cleanText = re.sub('#\S+\s', ' ', cleanText)
        cleanText = re.sub('@\S+', '  ', cleanText)
        cleanText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', cleanText)
        cleanText = re.sub(r'[^\x00-\x7f]', ' ', cleanText)
        cleanText = re.sub('\s+', ' ', cleanText)
        return cleanText
```

```
[10] cleanResume("my #### $ #  # hello @ world access it @gmain.com")
```

Project Evidence (PP1)

```
...     ('tfidf', TfidfTransformer(use_idf=True)),
...     ('clf', MultinomialNB(alpha=.01)),
... ])
```

```
[16] text_clf.fit(X_train['Resume Summary'], list(y_train))
```




```
[17] y_pred = text_clf.predict(X_test['Resume Summary'].to_list())
```

```
[18] import pickle
     pickle.dump(text_clf, open("nlp_model.dat", "wb"))
```

```
[19] with open('nlp_model.dat' , 'rb') as f:
```

```
[10] cleanResume("my #### $ #  # hello @ world access it @gmail.com")
```

```
'my hello world a ess it '
```

```
[11] data['Resume Summary'] = data['Resume Summary'].apply(lambda x: cleanResume(x))
```

```
[12] X = data.drop('KPI Value', axis=1)
     y = data['KPI Value']
```

```
[13] from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

```
[14] from sklearn.feature_extraction.text import TfidfVectorizer
     tf = TfidfVectorizer()
```

```
[15] text_clf = Pipeline([
     ...     ('vect', CountVectorizer(analyzer="word", stop_words="english")),
     ...     ('tfidf', TfidfTransformer(use_idf=True)),
     ...     ('clf', MultinomialNB(alpha=.01)),
     ... ])
```

```
[16] text_clf.fit(X_train['Resume Summary'], list(y_train))
```

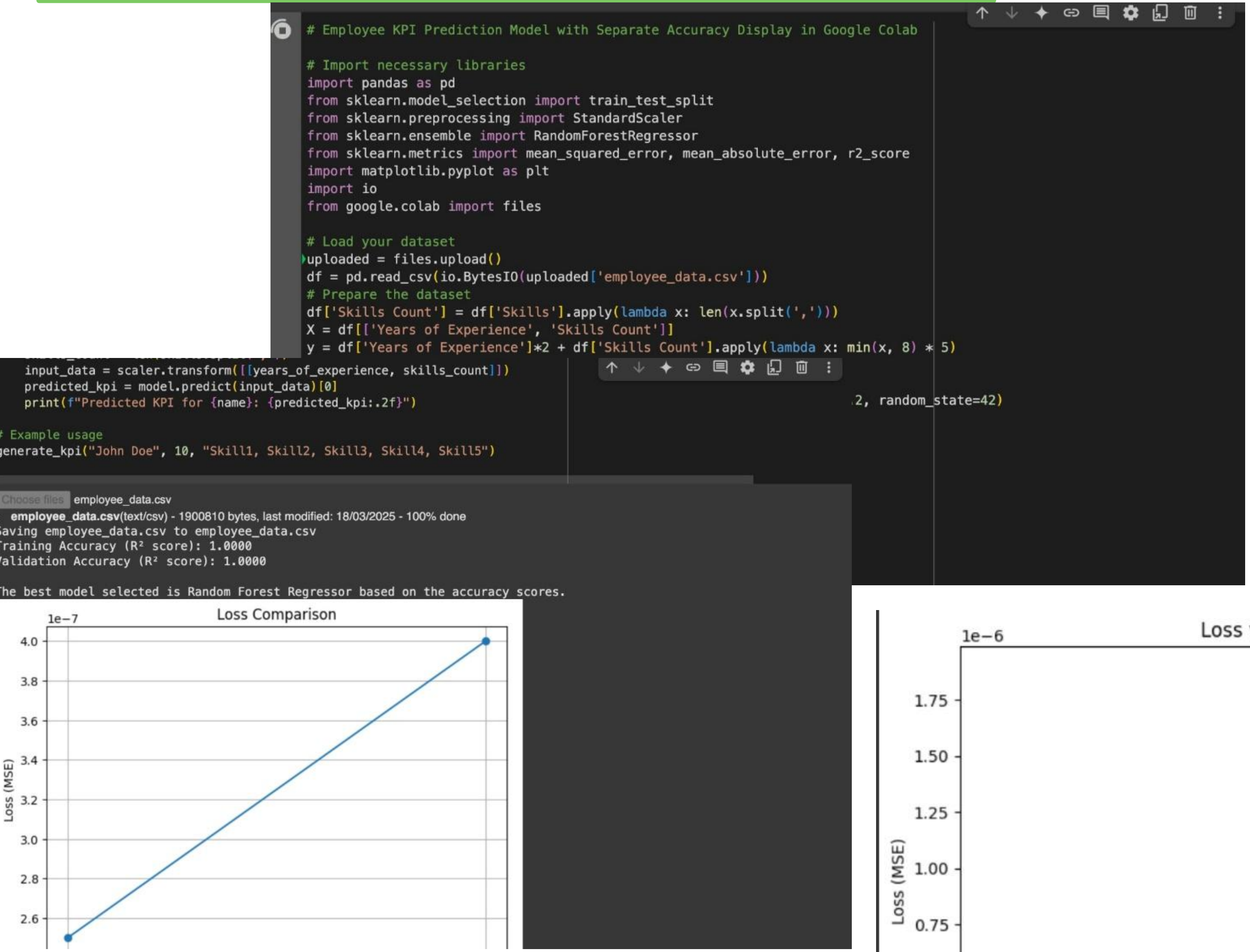
+ Code + Text

```
submit_button = widgets.Button(description="Submit")
submit_button.on_click(collect_data)

display(name_input, job_title_input, experience_input, skills_input, education_input, certifications_input, email_input, phone_number_input, loc
```


```
[260] Start coding or generate with AI.
```

Trained Models

Decision Tree Regression
Accuracy Score : 0.75



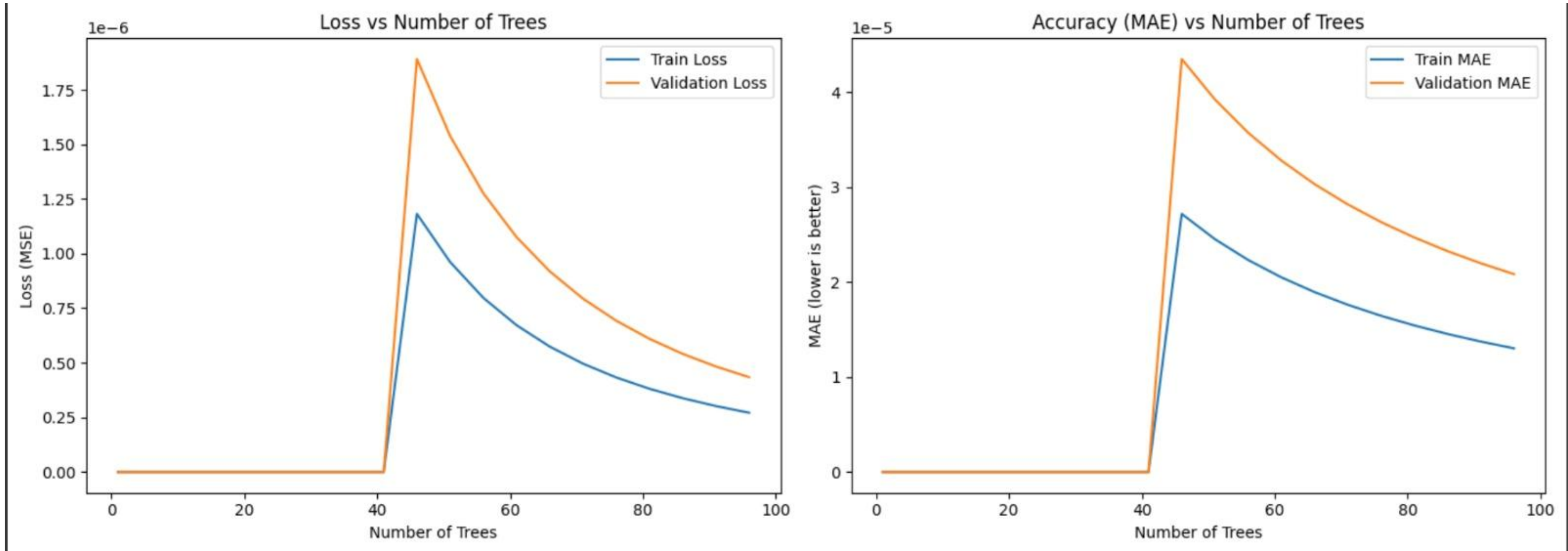
```
# Employee KPI Prediction Model with Separate Accuracy Display in Google Colab

# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
import io
from google.colab import files

# Load your dataset
uploaded = files.upload()
df = pd.read_csv(io.BytesIO(uploaded['employee_data.csv']))
# Prepare the dataset
df['Skills Count'] = df['Skills'].apply(lambda x: len(x.split(',')))
X = df[['Years of Experience', 'Skills Count']]
y = df['Years of Experience']*2 + df['Skills Count'].apply(lambda x: min(x, 8) * 5)
```

```
    input_data = scaler.transform([[years_of_experience, skills_count]])
    predicted_kpi = model.predict(input_data)[0]
    print(f"Predicted KPI for {name}: {predicted_kpi:.2f}")

# Example usage
generate_kpi("John Doe", 10, "Skill1, Skill2, Skill3, Skill4, Skill5")
```

```
2, random_state=42)
```

```
Choose files employee_data.csv
employee_data.csv(text/csv) - 1900810 bytes, last modified: 18/03/2025 - 100% done
Saving employee_data.csv to employee_data.csv
Training Accuracy (R² score): 1.0000
Validation Accuracy (R² score): 1.0000

The best model selected is Random Forest Regressor based on the accuracy scores.
```


Random Forest Regression
Accuracy Score:1

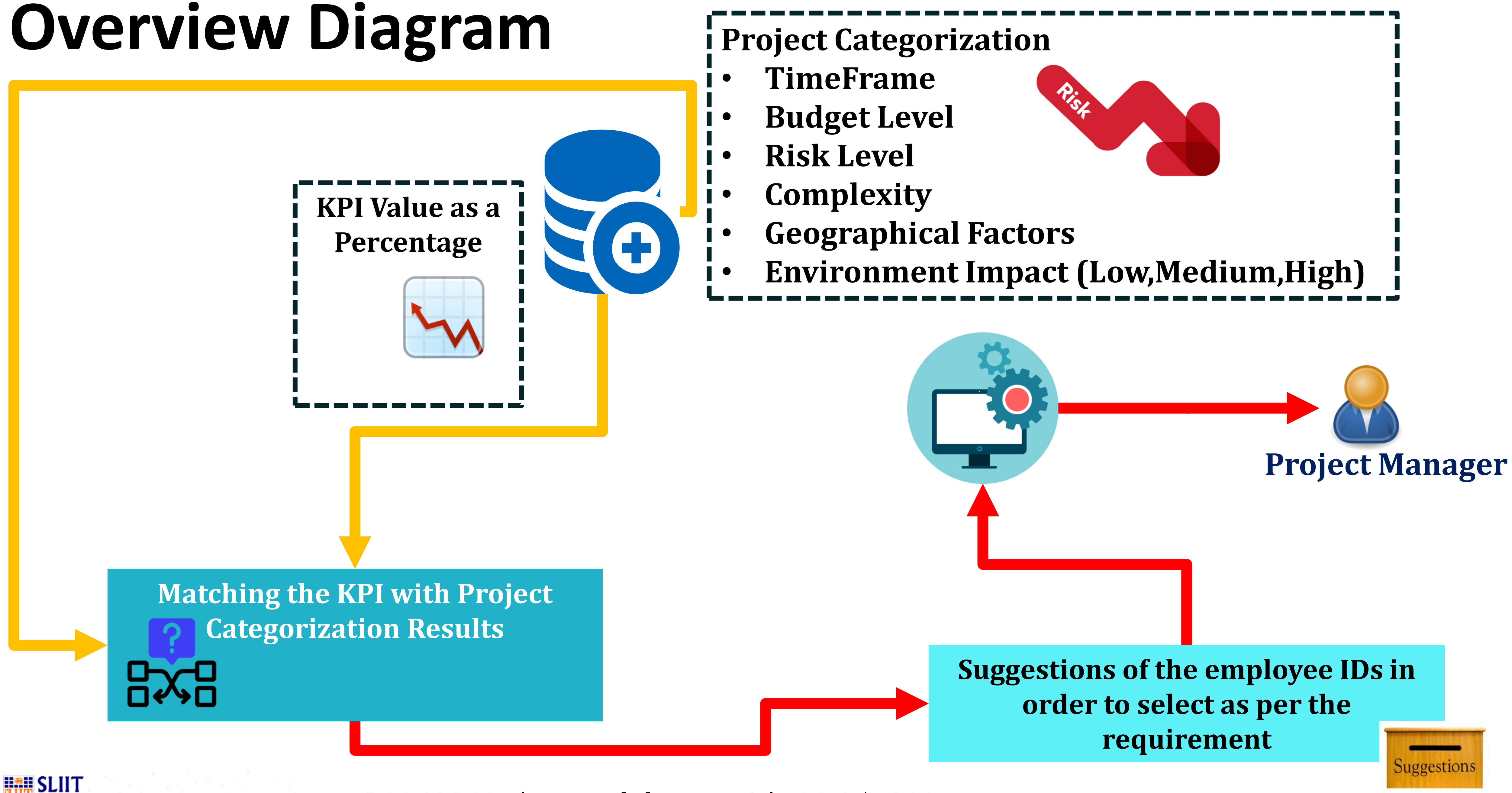


IT21069840|DEVASHIKA R.P.P.A

BSc(Hons)in Information Technology specialized in
Information Systems Engineering

Employee Allocation and Optimization.

Overview Diagram



Progress (100% Completion):

- Developed a model to match employees to projects using generated KPIs and project categorization results.
- Created suggestions for project managers on suitable employee selections.
- Validated allocation models with historical data and real-world scenarios.
- Implemented KPI percentage calculation for employee allocation.
- Used Decision Tree Regressor models and technologies like Python, ReactJS, NodeJS, and MySQL.
- Gathered data on number of employees working on projects and required employee categories.
- Application reference to tools like Procore, Primavera P6, BuildTrend, ALICE Technologies, and PlanGrid.

Importance & Industry Benefits:

- Improves efficiency by optimizing manpower allocation based on objective KPIs.
- Reduces inefficiencies caused by subjective allocation based on experience alone.
- Enhances project outcomes through better alignment of employee skills with project complexity.
- Supports systematic employee management and workload balancing.
- Facilitates data-driven decision-making in HR and project planning.

Project Evidence

```
# Define features (X) and target (y)
X = df.drop(columns=["KPI"])
y = df["KPI"]

# Split data into training (80%) and testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
print(X_train.shape,y_train.shape)#shape of training data set ..rows and columns
print(X_test.shape,y_test.shape)#shape of testing data set
```

```
Decision Tree Accuracy: 0.9750
Random Forest Accuracy: 0.9750
Gradient Boosting Accuracy: 0.9812
MLP Regressor Accuracy: 0.9875
Best Model: MLP Regressor with accuracy 0.9875
```

Plots Real KPI values in black (solid line). Plots Decision Tree predictions in blue (dashed line). Plots Random Forest predictions in green (dashed line). Plots Gradient Boosting predictions in red (dashed line). Uses the last 100 test samples for better visualization.

Risk Level: High

Complexity: Medium

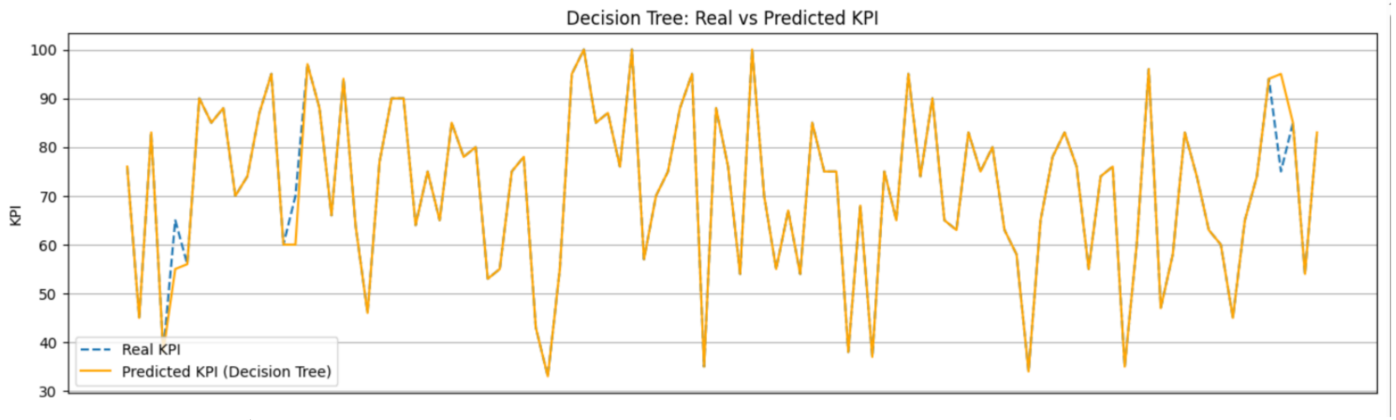
Budget Lev... Low

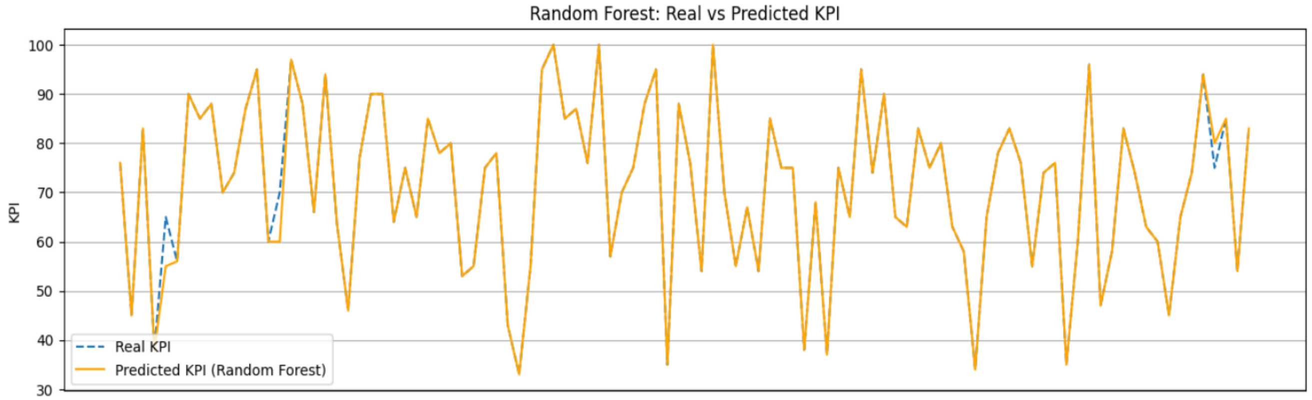
Job Title: Project Manager

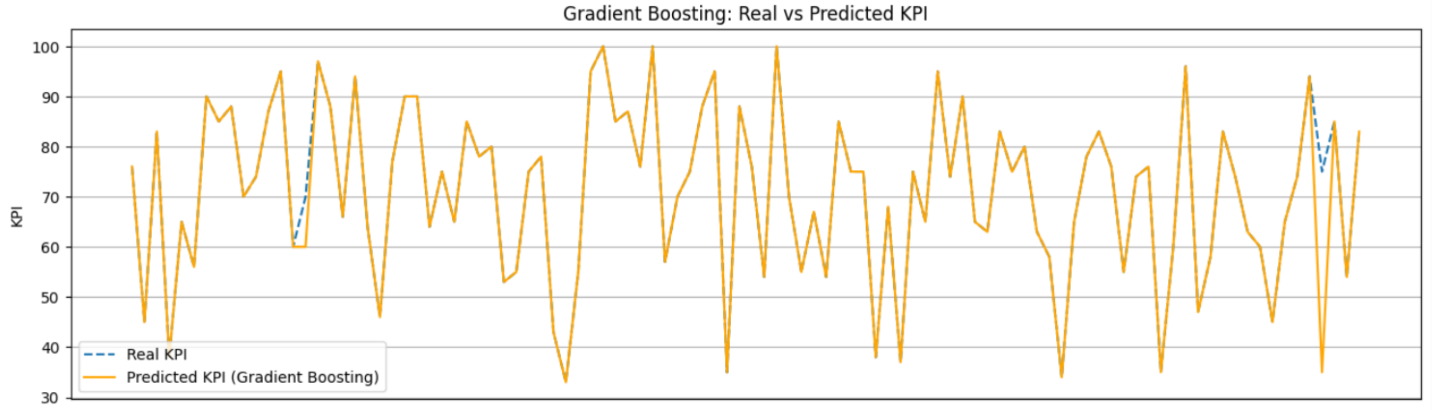
Submit

```
Predicted KPI with mlp_model: 66.17329014865373
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but MLPRegressor was fitted witł
  warnings.warn(
```

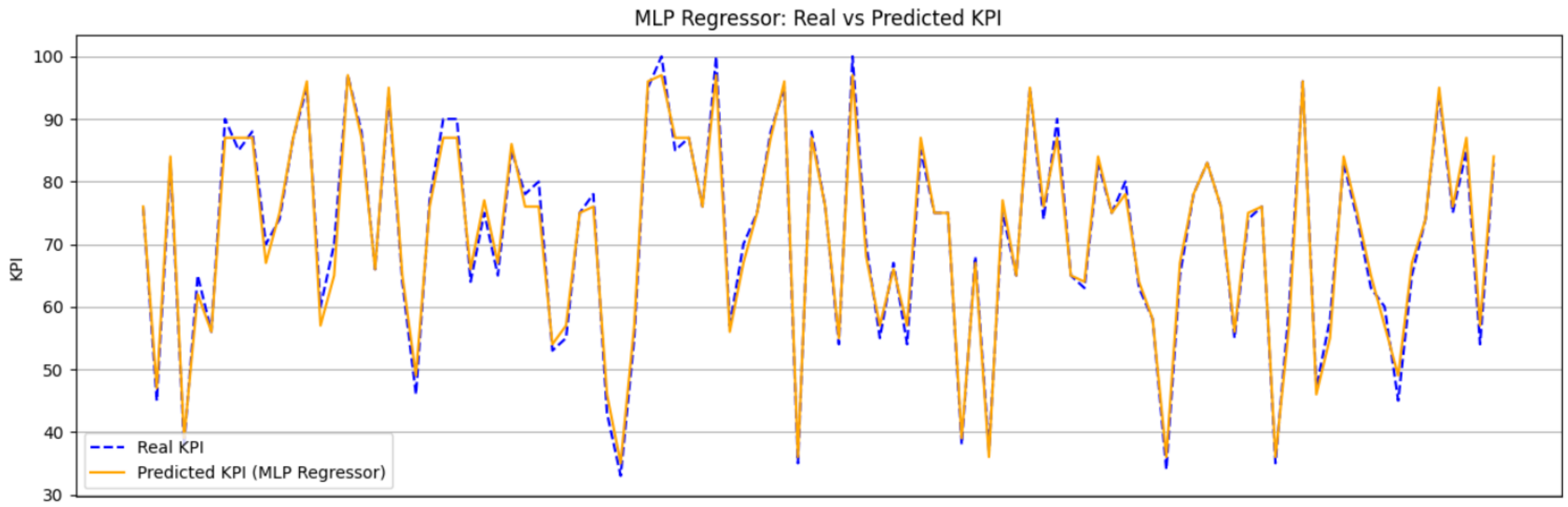
Project Evidence

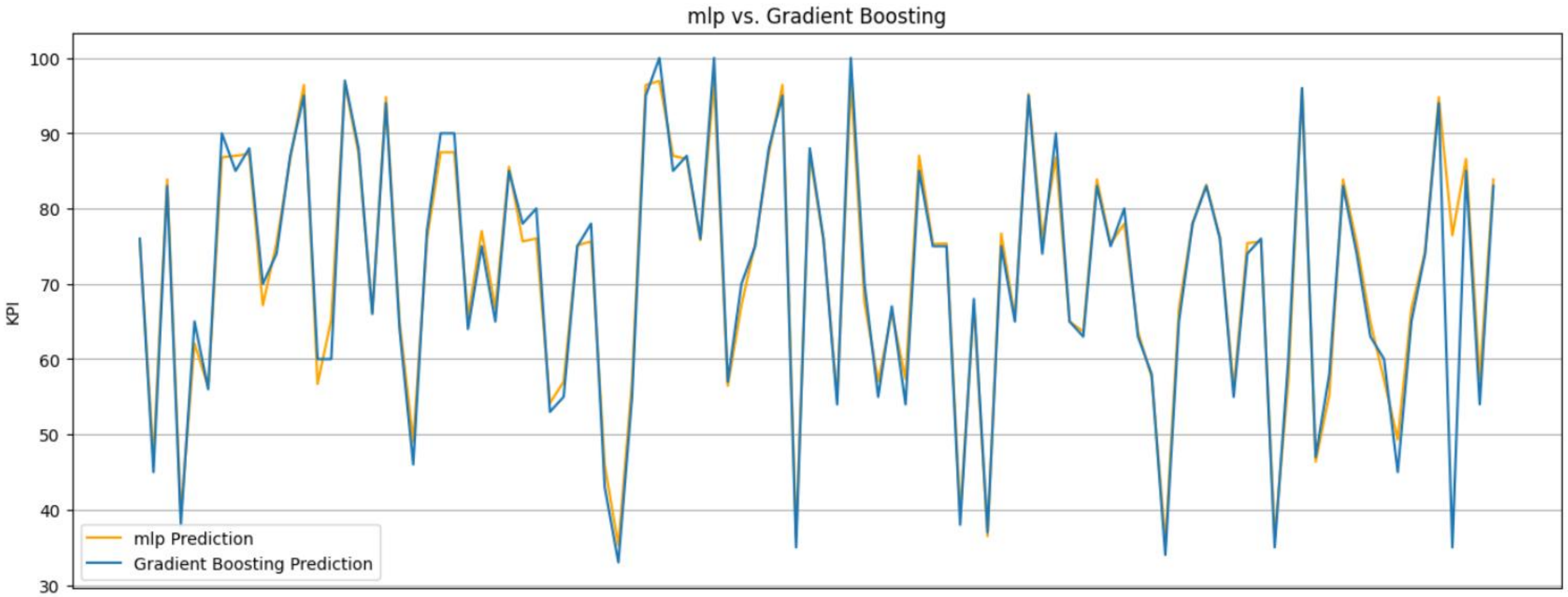






Project Evidence





IT21270956|MUNAGAMA M.K.H

BSc(Hons)in Information Technology specialized in Information Systems Engineering

Labor, cost and timeline prediction.

Overview Diagram

Project Manager

Recording the Daily Tasks and Labor Attendance

Enters Daily Attendance and Tasks

Predicting Cost and Timeline

Dashboard

Labor Prediction on Project Categorization

Project Categorization
- **TimeFrame**
- **Project Size**

Progress (100% Completion):

- Analyzed past project details to train labor, cost, and timeline prediction models.
- Created a platform to update daily labor attendance and completed tasks.
- Developed predictive models (Decision Tree Regression, Random Forest, Deep CNN) with accuracy scores for labor count and project timeline.
- Considered additional parameters like number of floors, windows, and doors in labor predictions.
- Integrated technologies like Python, ReactJS, NodeJS, MySQL, and Power BI for dashboards.
- Data collected from MAGA Pvt Ltd on budget, timeline, labor categories, and labor histograms.
- Provided real-time project status prediction capabilities.

Importance & Industry Benefits:

- Helps forecast labor requirements accurately, reducing under- or over-staffing.
- Enables better budgeting and timeline management, reducing project delays and cost overruns.
- Supports real-time monitoring of labor and task progress for proactive management.
- Enhances transparency and accountability in construction projects.
- Integrates with dashboard tools to provide actionable insights to project managers.

Project Evidence (PP1)

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

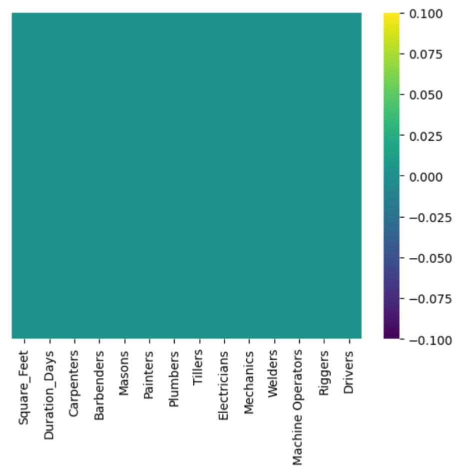
```
[2] data = pd.read_csv('Project details with labor count.csv', encoding='latin-1')
```

```
[3] data
```

	Project_ID	Square_Feet	Duration_Days	Carpenters	Barbenders	Masons	Painters	Plumbers	Tillers	Electricians	Mechanics	Welders	Machi Operato
0	1	15607	181	17	11	14	20	7	15	19	10	11	
1	2	19381	173	18	2	15	20	14	18	10	4	8	
2	3	14056	302	11	6	10	20	13	13	17	5	10	
3	4	5932	110	8	3	3	19	14	11	15	9	13	
4	5	5397	341	19	5	19	13	6	1	12	4	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	
95	96	19826	249	13	2	16	12	18	5	14	20	5	
96	97	9655	107	6	7	11	14	18	14	4	15	11	
97	98	10366	320	7	14	10	15	9	7	8	19	12	

0s completed at 04:08

[6] <Axes: >




```
[4] data = data.drop(columns=['Project_ID'])
```

```
[5] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Square_Feet        100 non-null    int64
 1   Duration_Days      100 non-null    int64
 2   Carpenters         100 non-null    int64
 3   Barbenders         100 non-null    int64
 4   Masons             100 non-null    int64
 5   Painters           100 non-null    int64
 6   Plumbers           100 non-null    int64
 7   Tillers            100 non-null    int64
 8   Electricians       100 non-null    int64
 9   Mechanics          100 non-null    int64
 10  Welders            100 non-null    int64
 11  Machine Operators  100 non-null    int64
 12  Riggers            100 non-null    int64
 13  Drivers            100 non-null    int64
dtypes: int64(14)
memory usage: 11.1 KB
```

```
[6] sns.heatmap(data.isnull(),yticklabels=False, cmap="viridis")
```

```
[7] y = data.drop(['Square_Feet','Duration_Days'], axis=1)
    X = data[['Square_Feet', 'Duration_Days']]
```

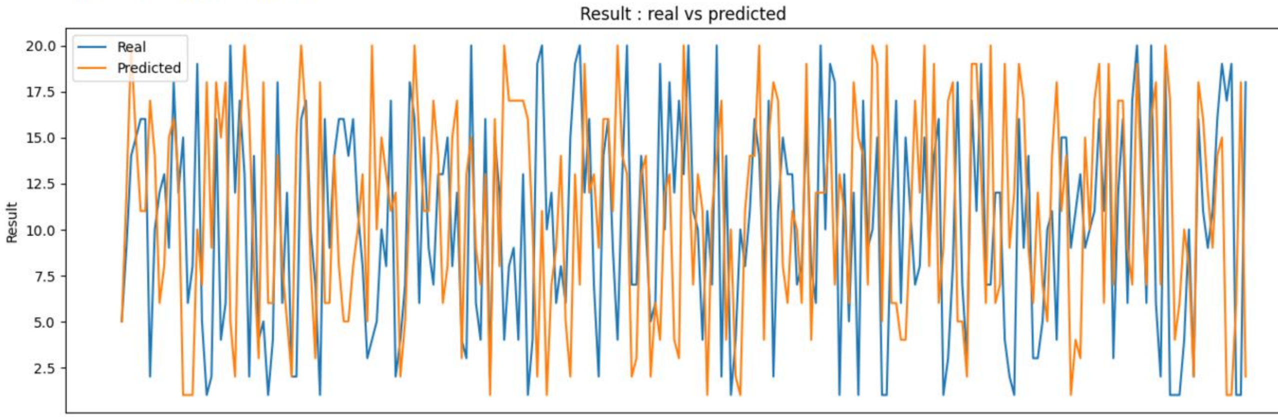
```
[8] from sklearn.svm import SVR
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import mean_squared_error

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=30)
```

```
[9] def model_score(model):
        model.fit(X_train, y_train)
        acc = model.score(X_test, y_test)
        print(str(model)+ ' | ' +str(acc))
```

```
[10] from sklearn.linear_model import LinearRegression
     lr = LinearRegression()
     model_score(lr)

     from sklearn.ensemble import RandomForestRegressor
     rf = RandomForestRegressor()
     model_score(rf)

     from sklearn.tree import DecisionTreeRegressor
     dt = DecisionTreeRegressor()
```

Project Evidence (PP1)

```
<class 'numpy.ndarray'>
<class 'pandas.core.frame.DataFrame'>
```




```
[16] def predict_workers():

        square_feet = float(input("Enter the Square Feet of the construction area: "))
        duration_days = int(input("Enter the Duration of the construction in days: "))

        sample_input = [[square_feet, duration_days]]

        predicted_workers = model.predict(sample_input)

        worker_columns = ['Carpenters', 'Barbenders', 'Masons', 'Painters', 'Plumbers', 'Tillers',
                          'Electricians', 'Mechanics', 'Welders', 'Machine Operators', 'Riggers', 'Drivers']

        predicted_workers = predicted_workers.round(0).astype(int)

        print("\nPredicted number of workers required:")
        for worker, count in zip(worker_columns, predicted_workers[0]):
            print(f"{worker}: {count}")
```

```
predict_workers()
```

```
Enter the Square Feet of the construction area: 2345
Enter the Duration of the construction in days: 1080

Predicted number of workers required:
Carpenters: 5
Barbenders: 3
Masons: 17
Painters: 4
Plumbers: 9
Tillers: 15
Electricians: 12
Mechanics: 2
Welders: 6
Machine Operators: 3
Riggers: 14
Drivers: 20
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning: X does not have valid feature name
```

Trained Models

Deep Convolutional Neural Network

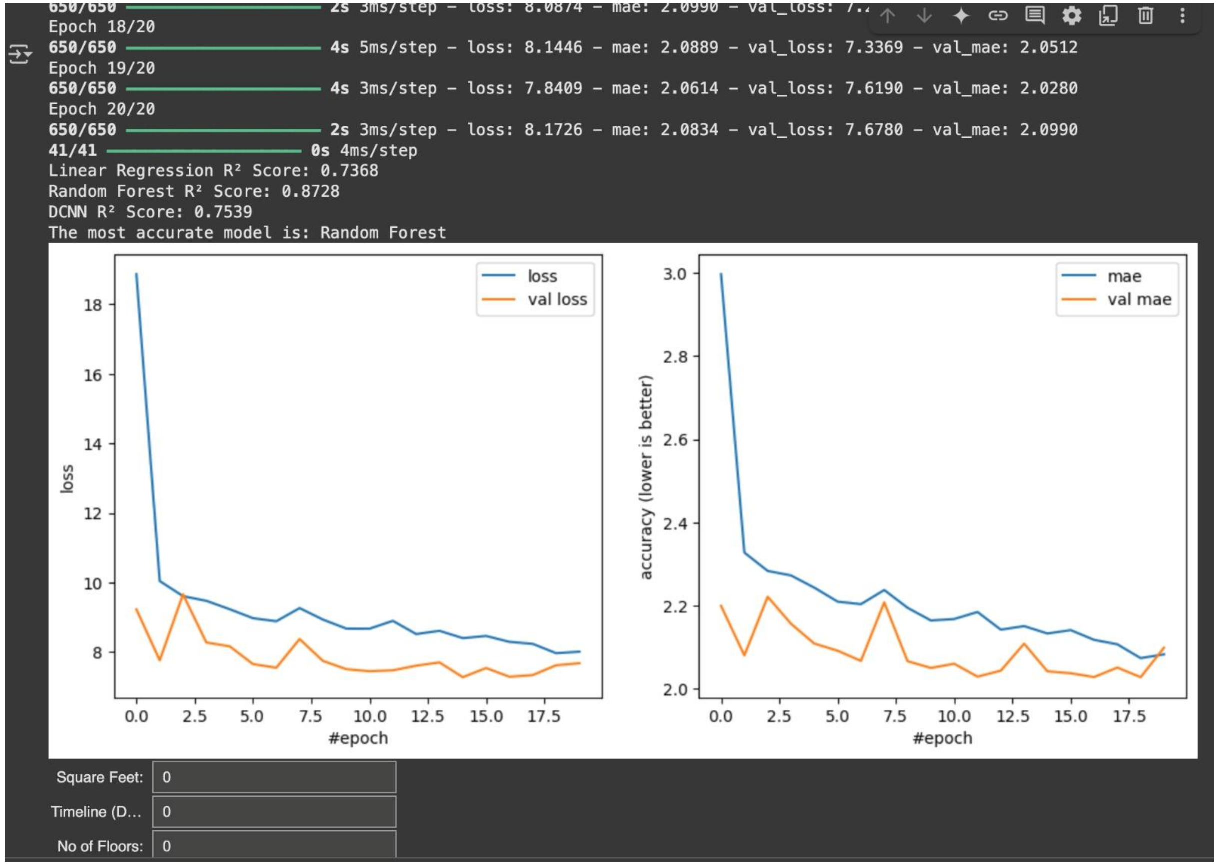
Accuracy Score : 0.7539

Linear Regression

Accuracy Score : 0.7368

Random Forest Regression

Accuracy Score : 0.8728



	Labor Type	Best Model Prediction
0	Carpenters	35.000000
1	Barbenders	23.000000
2	Masons	28.000000
3	Painters	19.000000
4	Plumbers	15.000000
5	Tillers	13.000000
6	Electricians	19.000000
7	Mechanics	11.000000
8	Welders	13.000000
9	Machine Operators	15.000000
10	Riggers	11.000000
11	Drivers	23.000000

Square Feet:	22427
Timeline (D...	840
No of Floors:	11
No of Wind...	6
No of Doors:	10

Predict Labor Count

```
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
          Labor Type  Best Model Prediction
```

Commercialization and Entrepenuership

- **Data Quality and Availability:** Access to comprehensive, accurate, and updated project and employee performance data remains a challenge.
- **Integration Complexity:** Seamless integration with diverse existing construction and HR management software requires significant customization.
- **User Adoption:** Resistance from traditional project managers and HR personnel accustomed to manual processes could slow adoption.
- **Scalability Issues:** Scaling the solution to accommodate very large projects or multinational companies may require advanced infrastructure.
- **Dynamic Project Environments:** Construction projects often change rapidly, requiring the system to adapt in near real-time.
- **Model Accuracy:** Maintaining and improving AI model accuracy with evolving data and project types needs ongoing training and validation.
- **Security and Privacy:** Handling sensitive employee and project data necessitates strong security protocols and compliance with regulations.
- **Cost of Implementation:** High upfront costs and training requirements could be a barrier for smaller companies.

Feedbacks from User Acceptance Testing

- UAT Done with our external Supervior **Mr.Darshana Senevirathna (Project Manager)** of **MAGA Enginnering Pvt Ltd.**

Click Here

Demonstration


Q&A

Thank You