



Machine Learning Approach For Employee Performance Prediction With IBM

Prepared by
Amit Kumar Thakur

CONTENT

1. Introduction.....	3
2. Objective.....	3
3. Project Description.....	3
4. Literature Survey.....	3
5. Theoretical Analysis.....	4
6. Experimental Investigation.....	5
7. Flow Chart.....	5
8. Result.....	6
9. Advantages.....	8
10. Disadvantages.....	9
11. Application.....	9
12. Conclusion.....	9
13. Appendix.....	10

INTRODUCTION :

L Overview:-

In this project we are going to analyze and predict the performance of employees in an organization on the basis of various factors, including, but not limited to, individual and domain specific characteristics, nature and level of schooling, socioeconomic status and different psychological factors.

L Purpose:-

The purpose of this project is to predict the performance of employees.

OBJECTIVE:

- Know fundamental concepts and techniques used for machine learning.
- Gain a broad understanding about data.
- Have knowledge on pre-processing the data/transformation techniques and some visualization concepts

PROJECT DESCRIPTION :

The Machine Learning Approach for Employee Performance Prediction with a comprehensive system designed to analyze various data points related to employees' work performance and use machine learning algorithms, leveraging ML technology stack, to predict and evaluate their future performance. By incorporating factors such as past performance metrics, training data, feedback, and external factors, the system aims to provide insights that can aid in talent management, resource allocation, and workforce optimization strategies.

LITERATURE SURVEY :

L Existing problem:

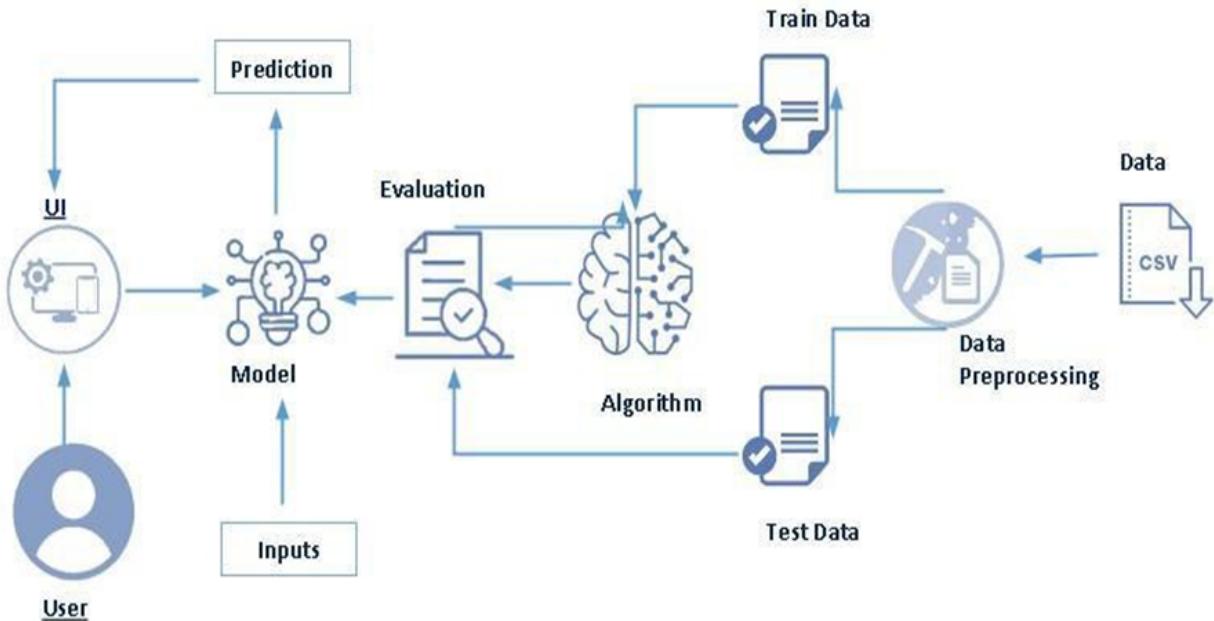
On previous system employee performance is calculated using paper works by evaluating the performance of the employee by hand

L Proposed solution:

As an alternative to the existing problem this project is made to automate the performance of the employee.

THEORETICAL ANALYSIS:

Block Diagram19



Hardware Minimum Requirement:

- CPU : PENTIUM III Processor
- Memory : 128 MB Cache : 512KB
- Floppy Disk : 1.44MB Hard Disk : 4.3GB
- Display : 15" Monitor
- Key Board: Standard 108 keys Enhanced Keyboard
- Mouse : MS Serial Mouse

Software Minimum Requirement:

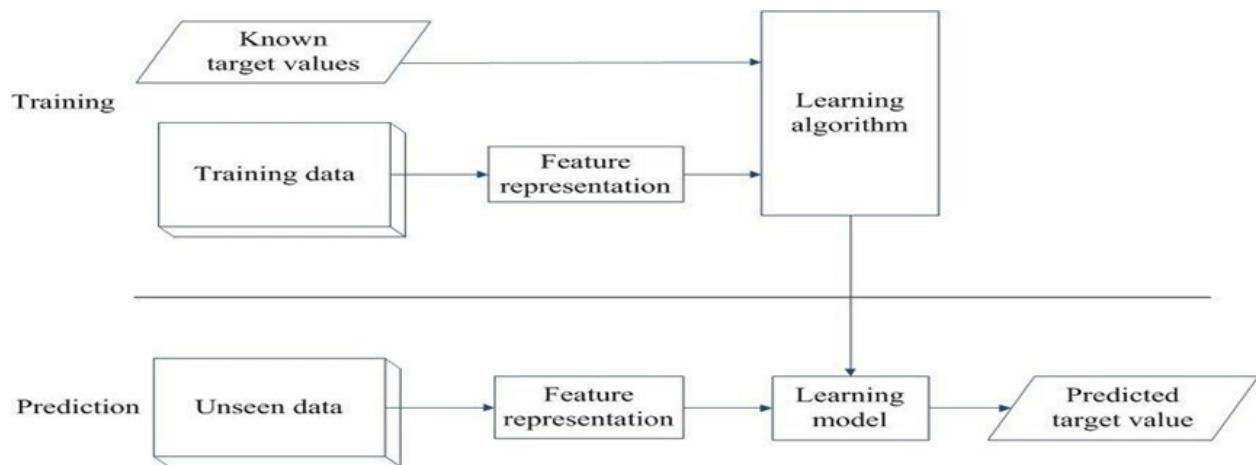
- Operating System : Windows XP, 7, 8 or above
- Front Tool : PHP
- Back End Tool : HTML

#EXPERIMENTAL INVESTIGATIONS :

Based on my analysis since the project is used with Supervised learning techniques namely Support Vector Machines, Random Forest, Naive Bayes, Neural Networks and Logistic Regression. The performance of the employee is analyzed based on the number of days the employee works ,target productivity acquired, over time they worked, how many team members etc and the most accurate result is found out

FLOW CHART :

1. User interacts with the UI to enter the input.
2. Entered input is analyzed by the model which is integrated.
3. Once model analyzes the input the prediction is showcased on the UI



RESULT :

[HOMEPAGE](#) :



> INPUT-1

Employee Performance Prediction Using Machine Learning

Home About Predict

quarter	7	department	4
day	61	team	7
targeted_productivity	0.80	smv	5.2
over_time	1	incentive	7
idle_time	2.2	idle_men	7
no_of_style_change	7	no_of_workers	7.5
month	5		

SUBMIT

>> OUTPUT-1:

The Employee is medium productive

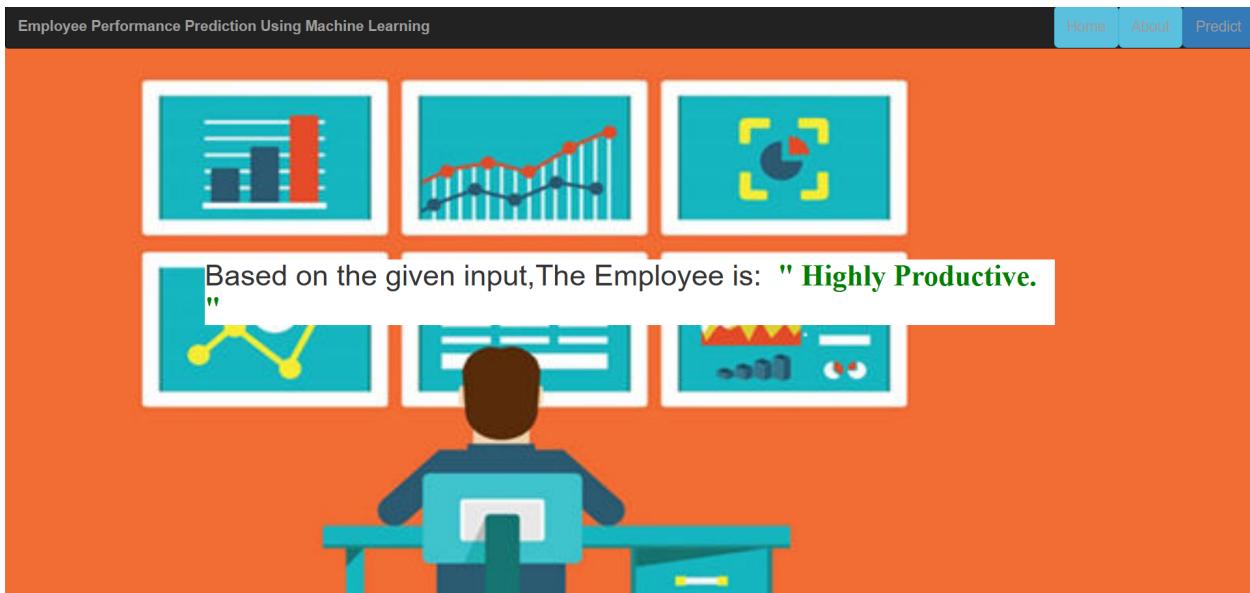


> INPUT-2:

A screenshot of the same web application, now showing the input form. The URL in the address bar is 127.0.0.1:5000/pred. The page has a purple background with a cartoon illustration of a person holding a tablet and another person working at a desk. The input fields are arranged in two columns. The left column includes: quarter (7), day (61), targeted_productivity (0.80), over_time (1), idle_time (2.2), no_of_style_change (7), and month (5). The right column includes: department (11), team (12), smv (5.2), incentive (7), idle_men (7), and no_of_workers (7.5). A green "SUBMIT" button is located at the bottom of the form.

>> OUTPUT-2:

The employee is Highly productive



ADVANTAGES:

1. Provides clarity
2. Enhances efficiency
3. Promotes job satisfaction
4. Increases motivation
5. Enables objective decision-making
6. Helps plan for training needs.

DISADVANTAGES:

1. The absence of goal setting and defined milestones
2. Using performance management solely as a measurement tool
3. Establishing trust
4. Untrained managers
5. It's an annual activity

APPLICATIONS:

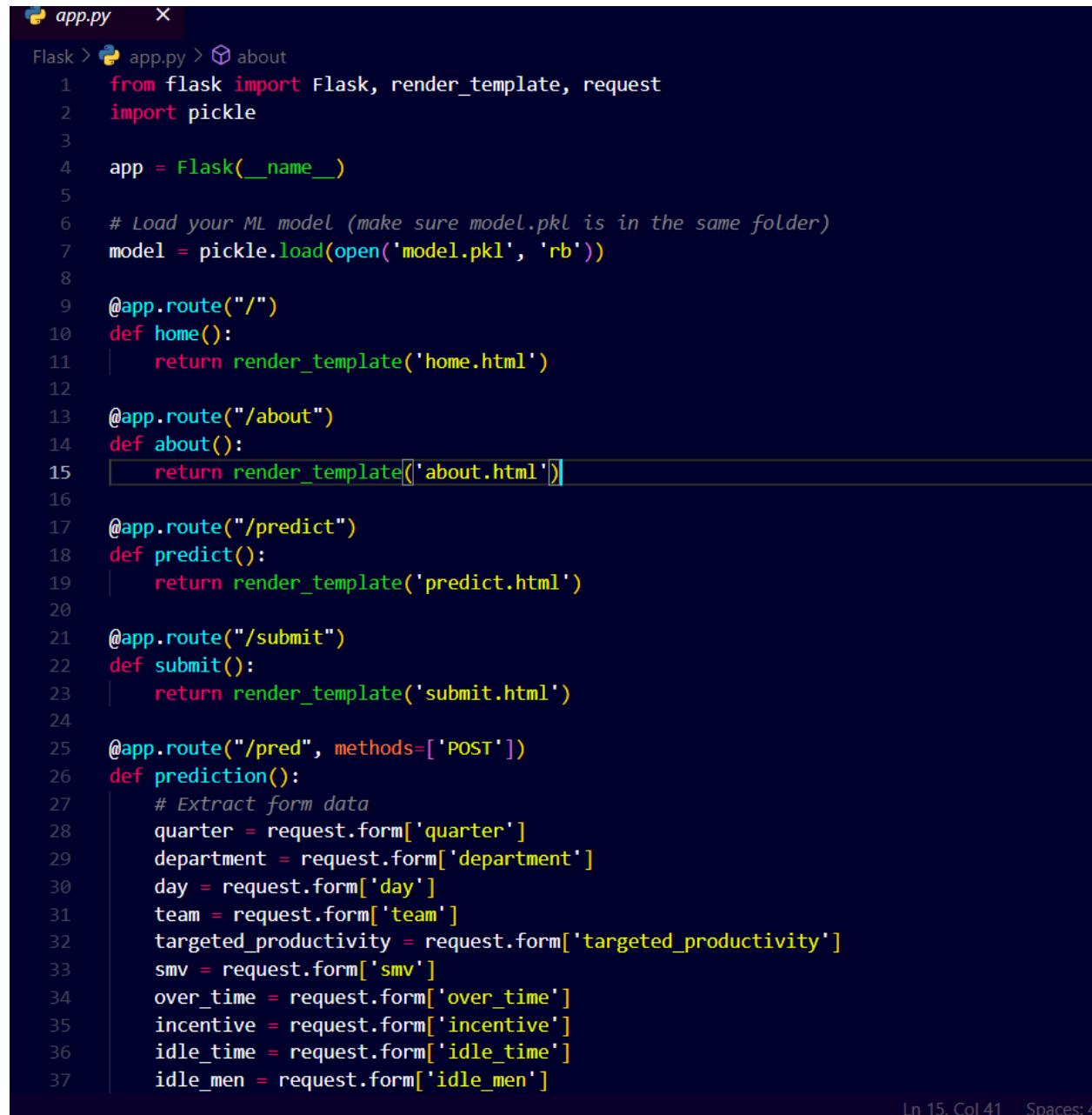
- a. Attendance
- b. Time management
- c. Training
- d. Initiative & innovation

CONCLUSION:

This project analyzes and predicts the performance of employees in an organization on the basis of various factors, including, but not limited to, individual and domain specific characteristics, nature and level of schooling, socioeconomic status and different psychological factors. The performance is evaluated successfully.

APPENDIX

1. app.py



The screenshot shows a code editor window with the file 'app.py' open. The code is written in Python using the Flask framework. It includes imports for Flask and pickle, defines a Flask application, loads an ML model from 'model.pkl', and defines routes for home, about, predict, submit, and prediction endpoints. The prediction endpoint handles POST requests and extracts form data to make a prediction using the loaded model.

```
Flask > app.py > about
1  from flask import Flask, render_template, request
2  import pickle
3
4  app = Flask(__name__)
5
6  # Load your ML model (make sure model.pkl is in the same folder)
7  model = pickle.load(open('model.pkl', 'rb'))
8
9  @app.route("/")
10 def home():
11     return render_template('home.html')
12
13 @app.route("/about")
14 def about():
15     return render_template('about.html')
16
17 @app.route("/predict")
18 def predict():
19     return render_template('predict.html')
20
21 @app.route("/submit")
22 def submit():
23     return render_template('submit.html')
24
25 @app.route("/pred", methods=['POST'])
26 def prediction():
27     # Extract form data
28     quarter = request.form['quarter']
29     department = request.form['department']
30     day = request.form['day']
31     team = request.form['team']
32     targeted_productivity = request.form['targeted_productivity']
33     smv = request.form['smv']
34     over_time = request.form['over_time']
35     incentive = request.form['incentive']
36     idle_time = request.form['idle_time']
37     idle_men = request.form['idle_men']
```

Ln 15, Col 41 Spaces: 4

```

37     idle_men = request.form['idle_men']
38     no_of_style_change = request.form['no_of_style_change']
39     no_of_workers = request.form['no_of_workers']
40     month = request.form['month']
41
42     # Prepare input for prediction - convert all to correct types
43     total = [[
44         int(quarter), int(department), int(day), int(team),
45         float(targeted_productivity), float(smv), int(over_time),
46         int(incentive), float(idle_time), int(idle_men),
47         int(no_of_style_change), float(no_of_workers), int(month)
48     ]]
49
50     prediction = model.predict(total)[0] # Assuming model.predict returns list/array
51
52     # Decide prediction message
53     if prediction <= 0.3:
54         text = 'Averagely Productive.'
55     elif 0.3 < prediction <= 0.8:
56         text = 'Medium Productive.'
57     else:
58         text = 'Highly Productive.'
59
60     # Render submit.html with prediction text
61     return render_template('submit.html', prediction_text=text)
62
63     if __name__ == "__main__":
64         app.run(debug=True)
65

```

2. home.html

```
home.html X
Flask > templates > home.html ...
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Home</title>
6     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" />
7     <style>
8       body {
9         background-image: url('{{ url_for('static', filename='h.png') }}');
10        background-size: cover;
11        background-color: lightgreen;
12      }
13    </style>
14  </head>
15  <body>
16    <nav class="navbar navbar-inverse">
17      <div class="container-fluid">
18        <div class="navbar-header">
19          <strong>
20            <a class="navbar-brand" href="{{ url_for('home') }}>
21              Employee Performance Prediction Using Machine Learning
22            </a>
23          </strong>
24        </div>
25        <ul class="nav navbar-nav navbar-right">
26          <li><a href="{{ url_for('home') }}" class="btn btn-info btn-lg">Home</a></li>
27          <li><a href="{{ url_for('about') }}" class="btn btn-info btn-lg">About</a></li>
28          <li><a href="{{ url_for('predict') }}" class="btn btn-primary btn-lg">Predict</a></li>
29        </ul>
30      </div>
31    </nav>
32    <center>
33      <!-- Your content here -->
34    </center>
35
36    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
37    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
38  </body>
```

3. about.html

```
5 about.html X
Flask > templates > 5 about.html > ⚏ html > ⚏ body > ⚏ div.container > ⚏ center
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8" />
6      <title>About</title>
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" />
8      <style>
9          body {
10              background: linear-gradient(to right, □rgba(0, 0, 0, 0.5), □rgba(0, 0, 0, 0.5)),
11                  url('{{ url_for('static', filename='h.png') }}');
12              background-size: cover;
13              background-color: ■powderblue;
14          }
15      </style>
16  </head>
17
18  <body>
19      <nav class="navbar navbar-inverse">
20          <div class="container-fluid">
21              <div class="navbar-header">
22                  <strong><a class="navbar-brand" href="{{ url_for('home') }}>Employee Performance Prediction Using Machine
23                      Learning</a></strong>
24              </div>
25              <ul class="nav navbar-nav navbar-right">
26                  <li><a href="{{ url_for('home') }}" class="btn btn-info btn-lg">Home</a></li>
27                  <li><a href="{{ url_for('about') }}" class="btn btn-info btn-lg">About</a></li>
28                  <li><a href="{{ url_for('predict') }}" class="btn btn-primary btn-lg">Predict</a></li>
29              </ul>
30          </div>
31      </nav>
32
33      <div class="container">
34          <center>
35              <h2>
36                  style="background: linear-gradient(to right, ■rgba(201, 201, 201, 0.6), ■rgba(201, 201, 201, 0.6));color:□black;margin-top:20%;margin-ri
37
38              <h2>
39                  style="background: linear-gradient(to right, ■rgba(201, 201, 201, 0.6), ■rgba(201, 201, 201, 0.6));color:□black;margin-top:20%;margin-ri
40
41                  Any business's success depends on its employees.<br />
42                  Businesses that realize this are concerned about employee output and productivity.<br />
43                  Productivity has a compounding effect at different levels in the workplace, meaning that high productivity at
44                  a lower level or vice versa can cascade.<br />
45                  Hence, analysis of performance of employees in any organization is the need of the hour.
46
47              </p>
48          </center>
49      </div>
50
51      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
52      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
53  </body>
54
55  </html>
```

4. submit.html

```
submit.html
Flask > templates > submit.html > html > body > div.container > h1 > span
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1" />
6      <title>Output</title>
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" />
8      <style>
9          body {
10              background-image: url("{{ url_for('static', filename='sub.png') }}");
11              background-size: cover;
12              background-color: #LightSteelBlue;
13          }
14          h1.big {
15              line-height: 1.8;
16          }
17      </style>
18  </head>
19  <body>
20
21  <nav class="navbar navbar-inverse">
22      <div class="container-fluid">
23          <div class="navbar-header">
24              <strong>
25                  <a class="navbar-brand" href="{{ url_for('home') }}>Employee Performance Prediction Using Machine Learning</a>
26              </strong>
27          </div>
28          <ul class="nav navbar-nav navbar-right">
29              <li><a href="{{ url_for('home') }}" class="btn btn-info btn-lg">Home</a></li>
30              <li><a href="{{ url_for('about') }}" class="btn btn-info btn-lg">About</a></li>
31              <li><a href="{{ url_for('predict') }}" class="btn btn-primary btn-lg">Predict</a></li>
32          </ul>
33      </div>
34  </nav>
35
36  <div class="container" style="width:70%">
37      <h1 style="background-color:#white; margin-top:23%; width:100%;">
38          Based on the given input,The Employee is:&nbsp;&nbsp;
39          <span style="color:#green;font-weight:bold;font-family:Times new roman;"> {{ prediction_text }} </span>
40      </h1>
41  </div>
42
43  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
44  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
45  </body>
46  </html>
47
```

5.predict.html

```
predict.html ✘
Flask > templates > predict.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Predict</title>
6      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
7      <style>
8          body {
9              background-image: url("{{ url_for('static', filename='Pred.png') }}");
10             background-size: cover;
11             background-color: #cadetblue;
12         }
13         h3.big {
14             line-height: 1.8;
15         }
16         .form-section {
17             background-color: #rgba(255, 255, 255, 0.8);
18             padding: 20px;
19             border-radius: 10px;
20             margin-top: 20px;
21         }
22     </style>
23 </head>
24 <body>
25
26 <nav class="navbar navbar-inverse">
27     <div class="container-fluid">
28         <div class="navbar-header">
29             <strong><a class="navbar-brand" href="{{ url_for('home') }}>Employee Performance Prediction</a></strong>
30         </div>
31         <ul class="nav navbar-nav navbar-right">
32             <li><a href="{{ url_for('home') }}" class="btn btn-info btn-lg">Home</a></li>
33             <li><a href="{{ url_for('about') }}" class="btn btn-info btn-lg">About</a></li>
34             <li><a href="{{ url_for('predict') }}" class="btn btn-primary btn-lg">Predict</a></li>
35         </ul>
36     </div>
37 </nav>
38
39 <div class="container form-section">
40     <form action="{{ url_for('prediction') }}" method="post">
41
42         <!-- Form fields (same as your original code) -->
43         <!-- Quarter & Department -->
44         <div class="form-group row">
45             <div class="col-xs-3">
46                 <label for="f1">Quarter</label>
47                 <input class="form-control" id="f1" name="quarter" required type="text">
48             </div>
49             <div class="col-xs-1"></div>
50             <div class="col-xs-3">
51                 <label for="f2">Department</label>
52                 <input class="form-control" id="f2" name="department" required type="text">
53             </div>
54         </div>
55
56         <!-- Day & Team -->
57         <div class="form-group row">
58             <div class="col-xs-3">
59                 <label for="f3">Day</label>
60                 <input class="form-control" id="f3" name="day" required type="text">
61             </div>
62             <div class="col-xs-1"></div>
63             <div class="col-xs-3">
64                 <label for="f4">Team</label>
65                 <input class="form-control" id="f4" name="team" required type="text">
66             </div>
67         </div>
68     </form>
```

```

69      <!-- Targeted Productivity & SMV -->
70      <div class="form-group row">
71          <div class="col-xs-3">
72              <label for="f5">Targeted Productivity</label>
73              <input class="form-control" id="f5" name="targeted_productivity" required type="text">
74          </div>
75          <div class="col-xs-1"></div>
76          <div class="col-xs-3">
77              <label for="f6">SMV</label>
78              <input class="form-control" id="f6" name="smv" required type="text">
79          </div>
80      </div>
81
82      <!-- Overtime & Incentive -->
83      <div class="form-group row">
84          <div class="col-xs-3">
85              <label for="f7">Over Time</label>
86              <input class="form-control" id="f7" name="over_time" required type="text">
87          </div>
88          <div class="col-xs-1"></div>
89          <div class="col-xs-3">
90              <label for="f8">Incentive</label>
91              <input class="form-control" id="f8" name="incentive" required type="text">
92          </div>
93      </div>
94
95      <!-- Idle Time & Idle Men -->
96      <div class="form-group row">
97          <div class="col-xs-3">
98              <label for="f9">Idle Time</label>
99              <input class="form-control" id="f9" name="idle_time" required type="text">
100         </div>
101
102         <div class="col-xs-1"></div>
103         <div class="col-xs-3">
104             <label for="f10">Idle Men</label>
105             <input class="form-control" id="f10" name="idle_men" required type="text">
106         </div>
107     </div>
108
109     <!-- Style Change & Workers -->
110     <div class="form-group row">
111         <div class="col-xs-3">
112             <label for="f11">No. of Style Change</label>
113             <input class="form-control" id="f11" name="no_of_style_change" required type="text">
114         </div>
115         <div class="col-xs-1"></div>
116         <div class="col-xs-3">
117             <label for="f12">No. of Workers</label>
118             <input class="form-control" id="f12" name="no_of_workers" required type="text">
119         </div>
120     </div>
121
122     <!-- Month -->
123     <div class="form-group row">
124         <div class="col-xs-3">
125             <label for="f13">Month</label>
126             <input class="form-control" id="f13" name="month" required type="text">
127         </div>
128     </div>
129
130     <!--  Submit button -->
131     <div class="form-group">
132         <button type="submit" class="btn btn-success btn-lg">SUBMIT</button>
133     </div>

```

Lp 1 Col 1 Spec

```
125 |     |     | <input class="form-control" id="t13" name="month" required type="text">
126 |     |     | </div>
127 |     |     | </div>
128 |
129 |     |     | <!-- ✅ Submit button -->
130 |     |     | <div class="form-group">
131 |     |     |     | <button type="submit" class="btn btn-success btn-lg">SUBMIT</button>
132 |     |     | </div>
133 |
134 |     | </form>
135 | </div>
136 |
137 | <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
138 | <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
139 | </body>
140 | </html>
141
```

6. model training:

jupyter Employee_performance_prediction Last Checkpoint: 8 days ago

File Edit View Run Kernel Settings Help Trusted

⊕ + × □ ▶ ■ ⌂ ▶ Markdown ▾

JupyterLab Python 3 (ipykernel) ▾

##importing library

```
[1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import MulticolumnLabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
import xgboost as Xgb
import pickle
```

##Read The Data

```
[2]: data=pd.read_csv("garments_worker_productivity.csv")
```

```
[3]: data.head()
```

	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	id
0	1/1/2015	Quarter1	sweing	Thursday	8		0.80	26.16	1108.0	7080	98	0.0	0	0	59.0
1	1/1/2015	Quarter1	finishing	Thursday	1		0.75	3.94	NaN	960	0	0.0	0	0	8.0
2	1/1/2015	Quarter1	sweing	Thursday	11		0.80	11.41	968.0	3660	50	0.0	0	0	30.5
3	1/1/2015	Quarter1	sweing	Thursday	12		0.80	11.41	968.0	3660	50	0.0	0	0	30.5
4	1/1/2015	Quarter1	sweing	Thursday	6		0.80	25.90	1170.0	1920	50	0.0	0	0	56.0



Linear Regression Model

```
[22]: from sklearn.linear_model import LinearRegression
model_lr = LinearRegression()

[23]: import pandas as pd

def preprocess_input_data(data_array):
    # Step 1: Define column names
    columns = ['Quarter', 'Department', 'Day', 'Feature1', 'Feature2', 'Feature3', 'Feature4',
               'Feature5', 'Feature6', 'Feature7', 'Feature8', 'Feature9', 'TargetCategory']

    # Step 2: Convert to DataFrame
    df = pd.DataFrame(data_array, columns=columns)

    # Step 3: Convert numeric columns
    numeric_cols = ['Feature1', 'Feature2', 'Feature3', 'Feature4',
                    'Feature5', 'Feature6', 'Feature7', 'Feature8', 'Feature9']
    df[numeric_cols] = df[numeric_cols].apply(pd.to_numeric)

    # Step 4: Encode categorical columns
    categorical_cols = ['Quarter', 'Department', 'Day']
    df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

    return df_encoded

[24]: x_train_encoded = preprocess_input_data(x_train)
model_lr.fit(x_train_encoded, y_train)

[24]: + LinearRegression
      LinearRegression()

      print("test_MSE:",mean_squared_error(y_test, pred))
      print("test_MAE:",mean_absolute_error(y_test, pred))
      print("R2_score:{}".format(r2_score(y_test, pred)))

      test_MSE: 0.011787060827356876
      test_MAE: 0.07284284213672713
      R2_score:0.6111674635852986

[36]: #XGBoost
pred = model_rf.predict(x_test_encoded)
print("test_MSE:",mean_squared_error(y_test, pred))
print("test_MAE:",mean_absolute_error(y_test, pred))
print("R2_score:{}".format(r2_score(y_test, pred)))

      test_MSE: 0.011787060827356876
      test_MAE: 0.07284284213672713
      R2_score:0.6111674635852986

Evaluating Performance Of The Model And Saving The Model

[37]: pred3 = model_xgb.predict(x_test_encoded)

[38]: print("test_MSE:",mean_squared_error(y_test, pred3))
print("test_MAE:",mean_absolute_error(y_test, pred3))
print("R2_score:{}".format(r2_score(y_test, pred3)))

      test_MSE: 0.00334252451720476
      test_MAE: 0.03372618245067457
      R2_score:0.8897365250685236

[40]: pickle.dump(model_xgb, open('model.pkl', 'wb'))
print(" Model successfully saved as 'model.pkl' ...by Amit_Kumar_Thakur")

      Model successfully saved as 'model.pkl' ...by Amit_Kumar_Thakur
```