

# **MACHINE LEARNING**

**(UML501)**

**End Semester Evaluation**

**PROJECT REPORT**

**ON**

**HR ANALYTICS**

**Submitted by: - Sahil Dhillon      (102003337)**

**Abhishek Gandhi (102003364)**

**Submitted to: - Ms. Suchita Sharma**



**THAPAR INSTITUTE**  
OF ENGINEERING & TECHNOLOGY  
(Deemed to be University)

**Computer Science and Engineering Department**

**TIET, Patiala**

**July 2022 – December 2022**

# CONTENTS

S.No.	Title	Page No.
1.	Introduction	3
2.	Detailed explanation of used technology	5
3.	Dataset Description	7
3.1	Code and Output	8
4.	Result	13
5.	References	14

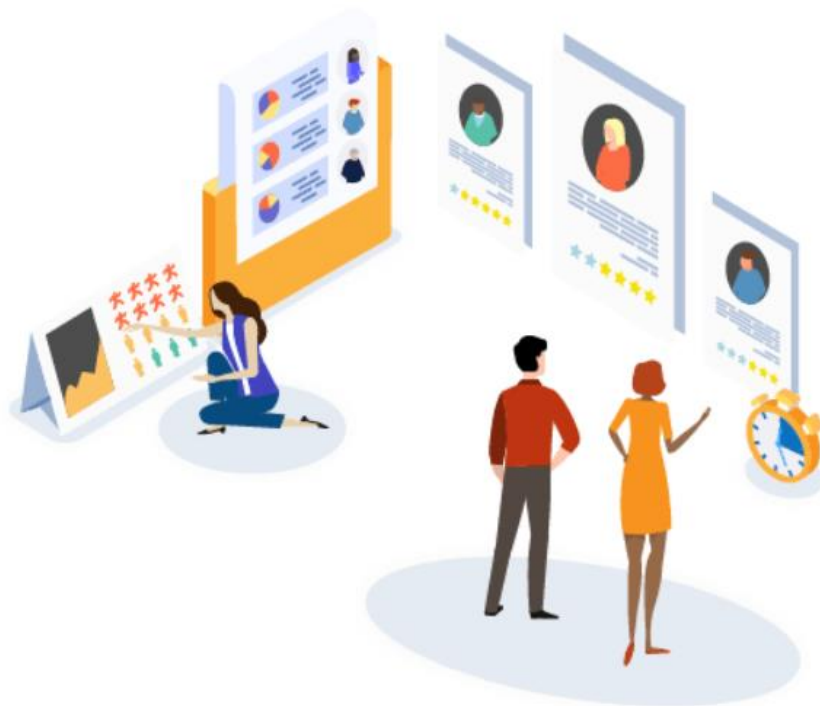
# INTRODUCTION

## What is HR Analytics?

HR analytics is the process of collecting and analysing Human Resource (HR) data in order to improve an organization's workforce performance. The process can also be referred to as talent analytics, people analytics, or even workforce analytics.

This method of data analysis takes data that is routinely collected by HR and correlates it to HR and organizational objectives. Doing so provides measured evidence of how HR initiatives are contributing to the organization's goals and strategies.

## Why is HR Analytics needed?



Most organizations already have data that is routinely collected, so why the need for a specialized form of analytics? Can HR not simply look at the data they already have?

Unfortunately, raw data on its own cannot actually provide any useful insight. It would be like looking at a large spreadsheet full of numbers and words. Without

organization or direction, the data appears meaningless. Once organized, compared and analysed, this raw data provides useful insight.

They can help answer questions like:

- What patterns can be revealed in employee turnover?
- How long does it take to hire employees?
- What amount of investment is needed to get employees up to a fully productive speed?
- **Which of our employees are most likely to leave within the year?**
- Are learning and development initiatives having an impact on employee performance?

## How does HR Analytics work?

**Analysis:** The analytical stage reviews the results from metric reporting to identify trends and patterns that may have an organizational impact. There are different analytical methods used, depending on the outcome desired. These include: descriptive analytics, prescriptive analytics, and predictive analytics.

Descriptive Analytics is focused solely on understanding historical data and what can be improved.

**Predictive Analytics uses statistical models to analyse historical data in order to forecast future risks or opportunities.**

Prescriptive Analytics takes Predictive Analytics a step further and predicts consequences for forecasted outcomes.

# TECHNOLOGY USED

## Python Libraries:

- **Numpy:** NumPy is a library for the Python programming language, using which we can add multi-dimensional arrays and matrices, complex mathematical formulation, etc.
- **Pandas:** mainly works with tabular data, also called SQL of ML. This included manipulating and analyzing data.
- **Matplotlib.pyplot:** plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object- oriented API for embedding plots into applications using general- purpose GUI toolkits like Tkinter, Qt, GTK.
- **Seaborn:** a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.
- **Pickle:** is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.
- **from sklearn.linear\_model import LogisticRegression:** LinearRegression fits a linear model with coefficients  $w = (w_1, \dots, w_p)$  to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.
- **from sklearn.ensemble import RandomForestClassifier:** A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples

of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree.

- **from sklearn import metrics:** Sklearn metrics are import metrics in SciKit Learn API to evaluate your machine learning algorithms.

## WEB FRAMEWORK

- **FLASK:** Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
- **HTML**
- **CSS**

# DATASET DESCRIPTION

RangeIndex: 14999 entries, 0 to 14998

Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	satisfaction_level	14999 non-null	float64
1	last_evaluation	14999 non-null	float64
2	number_project	14999 non-null	int64
3	average_monthly_hours	14999 non-null	int64
4	time_spend_company	14999 non-null	int64
5	work_accident	14999 non-null	int64
6	left	14999 non-null	int64
7	promotion_last_5years	14999 non-null	int64
8	Department	14999 non-null	object
9	salary	14999 non-null	object

dtypes: float64(2), int64(6), object(2)

# CODE AND OUTPUT

## HR Employee Prediction (i.e. whether they leave the company or continue to work)

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

### Loading Data

```
data=pd.read_csv('./hr-data.csv')
display(data)
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Department
0	0.38	0.53	2	157	3	0	1	0	sales
1	0.80	0.86	5	262	6	0	1	0	sales
2	0.11	0.88	7	272	4	0	1	0	sales
3	0.72	0.87	5	223	5	0	1	0	sales
4	0.37	0.52	2	159	3	0	1	0	sales
...	...	...	...	...	...	...	...	...	...
14994	0.40	0.57	2	151	3	0	1	0	support
14995	0.37	0.48	2	160	3	0	1	0	support
14996	0.37	0.53	2	143	3	0	1	0	support
14997	0.11	0.96	6	280	4	0	1	0	support
14998	0.37	0.52	2	158	3	0	1	0	support

14999 rows × 10 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype  
---  -
0   satisfaction_level    14999 non-null  float64
1   last_evaluation      14999 non-null  float64
2   number_project       14999 non-null  int64  
3   average_monthly_hours 14999 non-null  int64  
4   time_spend_company   14999 non-null  int64  
5   work_accident        14999 non-null  int64  
6   left                 14999 non-null  int64  
7   promotion_last_5years 14999 non-null  int64  
8   Department           14999 non-null  object  
9   salary               14999 non-null  object  
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

```
data.isnull().sum()
```

```
satisfaction_level    0
last_evaluation       0
number_project        0
average_monthly_hours 0
time_spend_company    0
work_accident         0
left                 0
promotion_last_5years 0
Department            0
salary               0
dtype: int64
```

### Select Left data from the column

```
left=data[data.left==1]
print("No. of employees left the company = ",left.shape[0])
```

No. of employees left the company = 3571



## Select Retained data from the column

```
retained=data[data.left==0]
print("No. of employees retained in the company = ",retained.shape[0])
```

No. of employees retained in the company = 11428

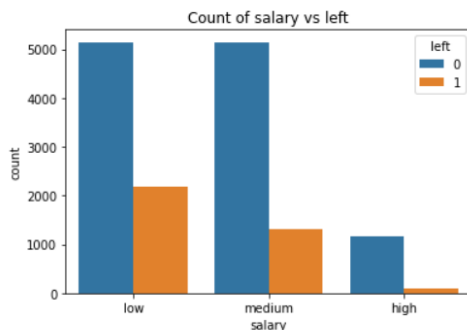
## Group Data on Left base value

```
data.groupby('left').mean()
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	promotion_last_5years
left							
0	0.666810	0.715473	3.786664	199.060203	3.380032	0.175009	0.026251
1	0.440098	0.718113	3.855503	207.419210	3.876505	0.047326	0.005321

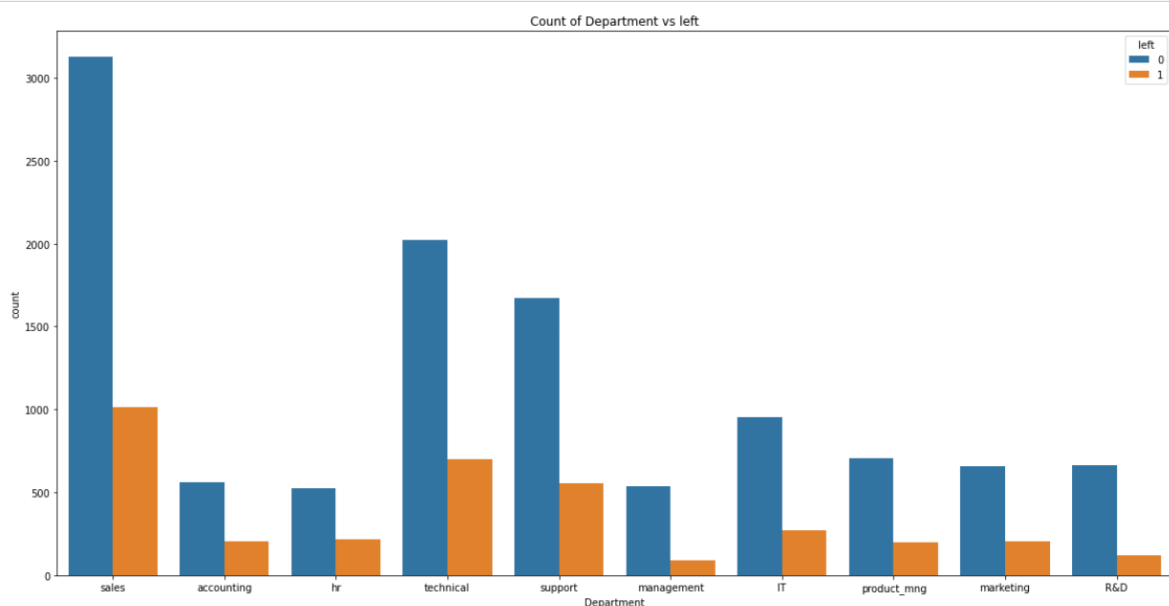
## Bar Chart Showing Impact of Employees Salaries on Retention

```
plt.title('Count of salary vs left')
sns.countplot(data=data, x='salary', hue='left');
```



## Bar Chart Showing Correlation between department and employee retention

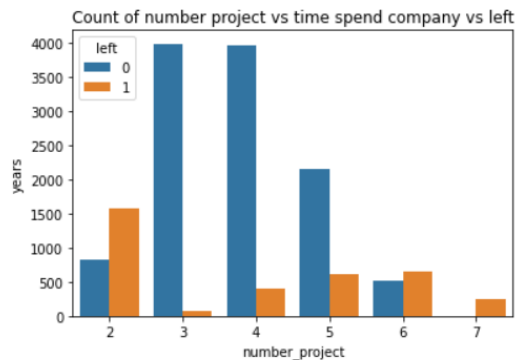
```
plt.figure(figsize=(20,10))
plt.title('Count of Department vs left')
sns.countplot(data=data, x='Department', hue='left');
```



## Bar Chart Showing Correlation between Number of projects done for company and People Who Left

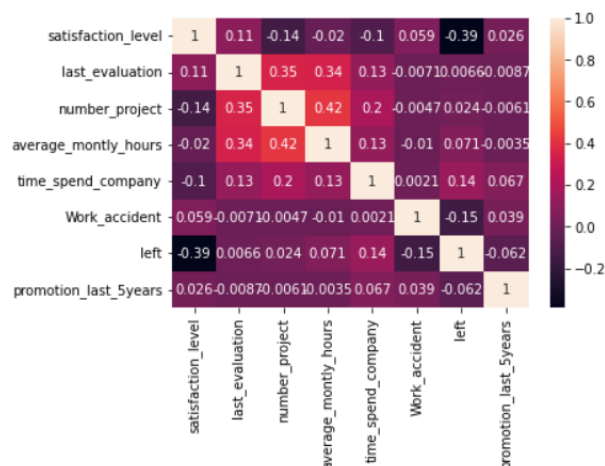
```
plt.title('Count of number project vs time spend company vs left')
c=sns.countplot(data=data, x='number_project', hue='left');
c.set_ylabel("years")
c
```

<AxesSubplot:title={'center':'Count of number project vs time spend company vs left'}, xlabel='number\_project', ylabel='years'>



## Correlation

```
sns.heatmap(data=data.corr(),annot=True);
```



## Selecting Variables Which Impact Most on Employee Retention

```
sel_data = data[['satisfaction_level', 'average_monthly_hours', 'time_spent_company', 'Work_accident', 'promotion_last_5years', 'salary']]
sel_data.head()
```

	satisfaction_level	average_monthly_hours	time_spent_company	Work_accident	promotion_last_5years	salary
0	0.38	157	3	0	0	low
1	0.80	262	6	0	0	medium
2	0.11	272	4	0	0	medium
3	0.72	223	5	0	0	low
4	0.37	159	3	0	0	low

## Encoding Salary Variable

```
dummies = pd.get_dummies(sel_data.salary)
data_encoded = pd.concat([sel_data,dummies],axis='columns')
data_encoded=data_encoded.drop(['salary'],axis=1)
```

## Assign Data to X

```
X = data_encoded
X.head()
```

	satisfaction_level	average_monthly_hours	time_spend_company	Work_accident	promotion_last_5years	high	low	medium
0	0.38	157	3	0	0	0	1	0
1	0.80	262	6	0	0	0	0	1
2	0.11	272	4	0	0	0	0	1
3	0.72	223	5	0	0	0	1	0
4	0.37	159	3	0	0	0	1	0

## Left data assign to y

```
y = data.left
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.3)
```

## Logistic Regression Model For Prediction

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
linear_model = LogisticRegression()
# creating a RF classifier
clf = RandomForestClassifier(n_estimators = 100)
```

## Train Model

```
linear_model.fit(X_train, y_train)
# Training the model on the training dataset
# fit function is used to train the model using the training sets as parameters
clf.fit(X_train, y_train)
```

RandomForestClassifier()

## Prediction

```
linear_model.predict(X_test)
# performing predictions on the test dataset
y_pred=clf.predict(X_test)
```

## Model Score

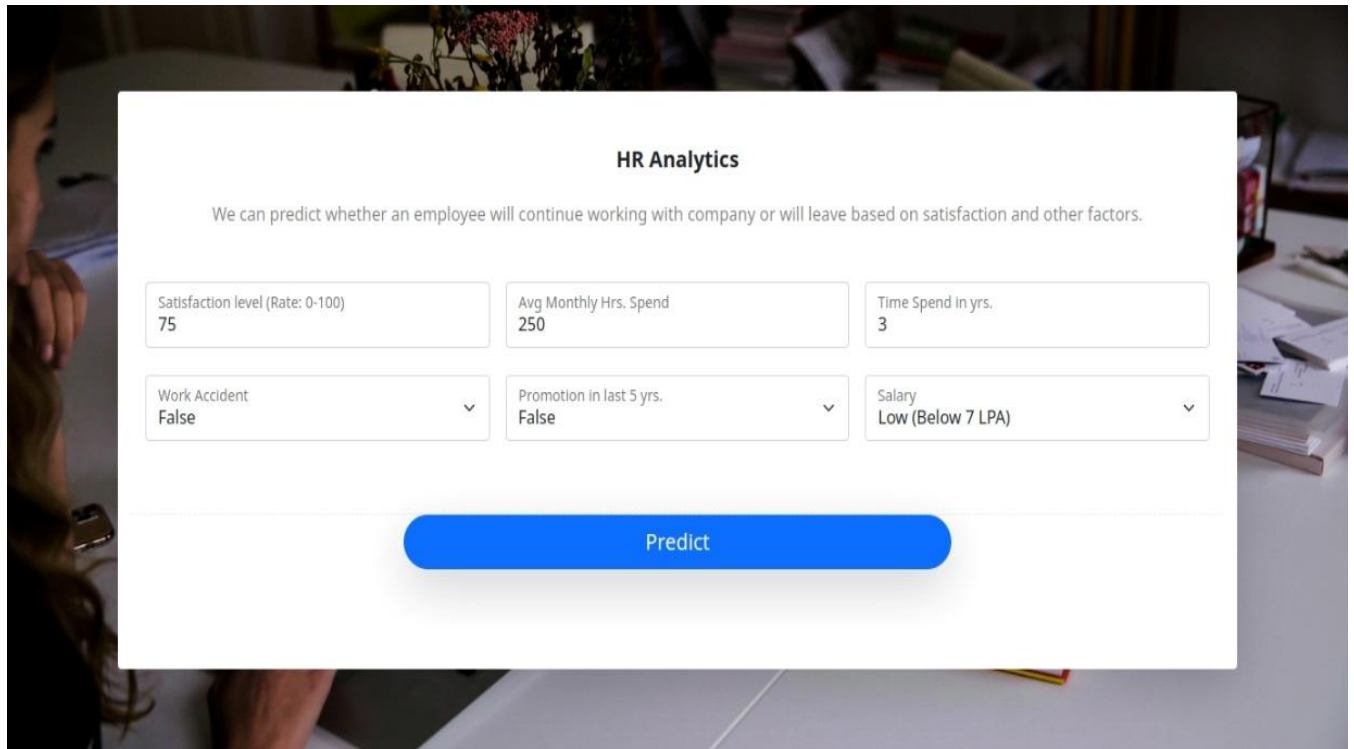
```
from sklearn import metrics
print("Accuracy Of The Model Using Linear Regression : ",linear_model.score(X_test,y_test))
print("ACCURACY OF THE MODEL Using Random Forest Classifier : ", metrics.accuracy_score(y_test, y_pred))
```

Accuracy Of The Model Using Linear Regression : 0.7662857142857142  
ACCURACY OF THE MODEL Using Random Forest Classifier : 0.9720952380952381

## Converting to Pickle model

```
pickle.dump(clf,open('hr_rf_model.pkl','wb'))
model=pickle.load(open('hr_rf_model.pkl','rb'))
```

# RESULTS

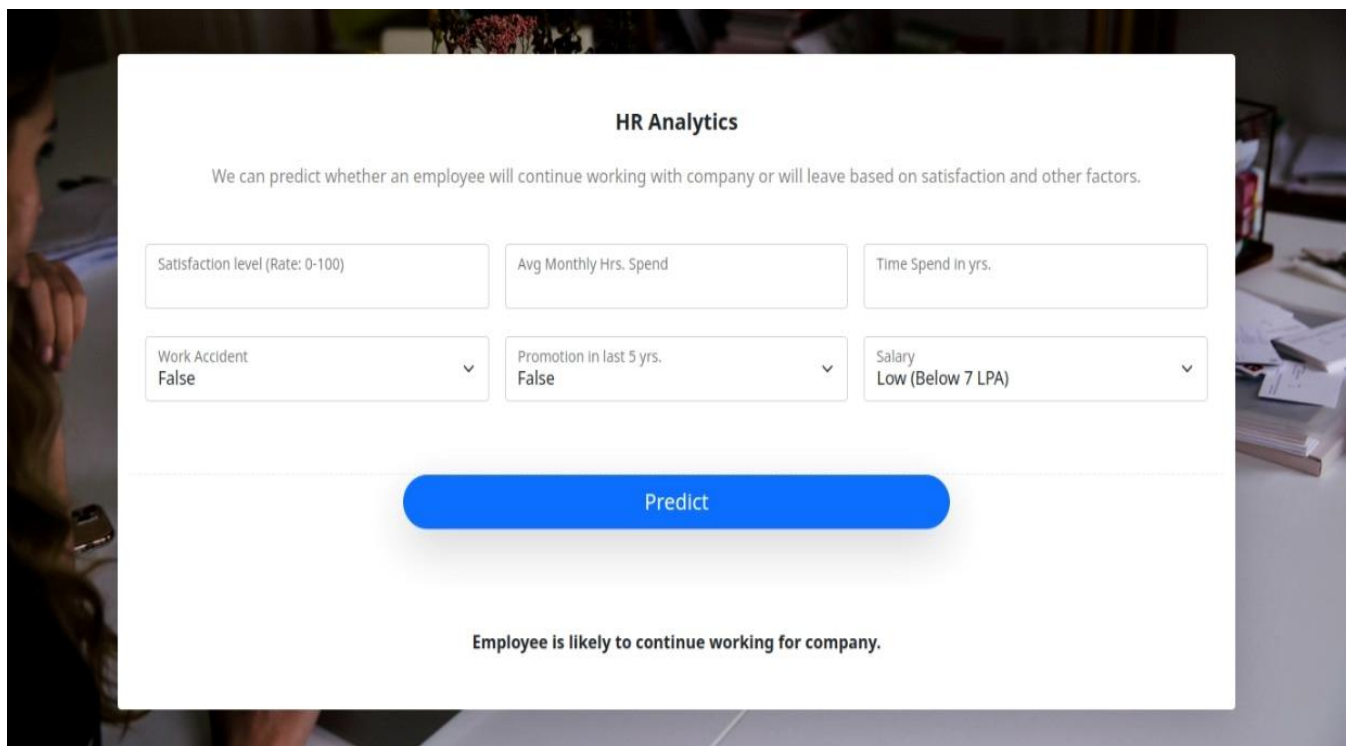


**HR Analytics**

We can predict whether an employee will continue working with company or will leave based on satisfaction and other factors.

Satisfaction level (Rate: 0-100) 75	Avg Monthly Hrs. Spend 250	Time Spend in yrs. 3
Work Accident False	Promotion in last 5 yrs. False	Salary Low (Below 7 LPA)

Predict



**HR Analytics**

We can predict whether an employee will continue working with company or will leave based on satisfaction and other factors.

Satisfaction level (Rate: 0-100)	Avg Monthly Hrs. Spend	Time Spend in yrs.
Work Accident False	Promotion in last 5 yrs. False	Salary Low (Below 7 LPA)

Predict

Employee is likely to continue working for company.

# REFERENCES

Dataset source: <https://www.kaggle.com/datasets/mfaisalqureshi/hr-analytics-and-job-prediction>

Project regarding Hr analytics: <https://www.kaggle.com/code/mfaisalqureshi/hr-analytics-logistics-regression>