

```
In [1]: import pandas as pd
import sklearn as sk
```

```
In [2]: ed=pd.read_csv('/content/sample_data/Employee data.csv')
```

```
In [3]: ed.head()
```

Out [3]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales
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

```
In [4]: ed.tail()
```

Out [4]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales
14994	0.40	0.57	2	151	3	0	1	0	sales
14995	0.37	0.48	2	160	3	0	1	0	sales
14996	0.37	0.53	2	143	3	0	1	0	sales
14997	0.11	0.96	6	280	4	0	1	0	sales
14998	0.37	0.52	2	158	3	0	1	0	sales

```
In [5]: ed.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   sales                       14999 non-null  object
9   salary                      14999 non-null  object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

```
In [6]: ed.describe()
```

Out [6]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	sales
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268	sales
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281	sales
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000	sales
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000	sales
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000	sales
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000	sales
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000	sales

```
In [7]: ed.shape
```

Out [7]: (14999, 10)

```
In [8]: categorical_cols = ed.select_dtypes(include=['object']).columns
print(categorical_cols)
```

```
Index(['sales', 'salary'], dtype='object')
```

```
In [10]: ed['sales'].unique()
```

Out [10]: array(['sales', 'accounting', 'hr', 'technical', 'support', 'management', 'IT', 'product_mng', 'marketing', 'RandD'], dtype=object)

```
In [14]: ed['sales']=ed['sales'].replace(['sales', 'accounting', 'hr', 'technical', 'support', 'management', 'IT', 'product_mng', 'marketing', 'RandD'], True)
pd.set_option('future.no_silent_downcasting', True)
```

```
In [15]: ed['sales'].value_counts(normalize=True)
```


	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	promotion_last_5years	sales
5191	0.52	0.96	4	246	3	0	0	5
13418	0.49	0.65	4	233	7	0	0	1
5390	0.66	0.73	5	249	2	0	0	5
860	0.79	1.00	4	218	5	0	0	1
7270	0.98	0.86	2	219	4	0	0	1

11999 rows × 9 columns

In [33]: X_test

Out [33]:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	promotion_last_5years	sales
6723	0.65	0.96	5	226	2	1	0	9
6473	0.88	0.80	3	166	2	0	0	7
4679	0.69	0.98	3	214	2	0	0	1
862	0.41	0.47	2	154	3	0	0	1
7286	0.87	0.76	5	254	2	1	0	3
...
3297	0.61	0.74	2	143	6	0	0	1
14113	0.93	0.89	3	255	7	1	0	1
5514	0.78	0.61	3	193	3	0	0	1
9939	0.89	0.76	2	278	2	0	0	1
14346	0.31	0.84	7	133	5	0	0	4

3000 rows × 9 columns

In [34]: y_train

Out [34]:

	left
9838	0
7689	0
6557	0
6872	0
820	1
...	...
5191	0
13418	0
5390	0
860	1
7270	0

11999 rows × 1 columns

dtype: int64

In [36]: y_test

Out [36]:

	left
6723	0
6473	0
4679	0
862	1
7286	0
...	...
3297	0
14113	0
5514	0
9939	0
14346	1

3000 rows × 1 columns

dtype: int64

```
In [37]: from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X_train,y_train)
```

```
Out [37]: RandomForestClassifier
RandomForestClassifier()
```

```
In [38]: y_pred=rf.predict(X_test)
accuracy_score(y_test,y_pred)
```

```
Out [38]: 0.9886666666666667
```

```
In [39]: rf=RandomForestClassifier()
rf.fit(X,y)
```

```
Out [39]: RandomForestClassifier
RandomForestClassifier()
```

```
In [40]: print(ed.columns)
```

```
Index(['satisfaction_level', 'last_evaluation', 'number_project',
       'average_monthly_hours', 'time_spend_company', 'Work_accident', 'left',
       'promotion_last_5years', 'sales', 'salary'],
      dtype='object')
```

```
In [42]: new_data = pd.DataFrame({'satisfaction_level':[0.36], 'last_evaluation':[0.54], 'number_project':[6],
       'average_monthly_hours':[265], 'time_spend_company':[5], 'Work_accident':[0], 'left':[0],
       'promotion_last_5years':[1], 'sales':[10], 'salary':[2]},index=[0])
```

```
In [43]: new_data_aligned = new_data[rf.feature_names_in_]
```

```
In [44]: p = rf.predict(new_data_aligned)
print("Prediction:", p)
```

```
Prediction: [0]
```

```
In [47]: from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
In [48]: import joblib
```

```
In [49]: joblib.dump(rf, '/content/drive/My Drive/Models/employee_turnover_model.pkl')
```

```
Out [49]: ['/content/drive/My Drive/Models/employee_turnover_model.pkl']
```

```
In [50]: loaded_model = joblib.load('/content/drive/My Drive/Models/employee_turnover_model.pkl')
```

```
In [52]: prediction = loaded_model.predict(new_data_aligned)
```

```
In [53]: print(prediction)
```

```
[0]
```

```
In [54]: print(type(loaded_model))
```

```
<class 'sklearn.ensemble._forest.RandomForestClassifier'>
```

```
In [56]: import os
```

```
model_path = '/content/drive/My Drive/models/employee_turnover_model.pkl'
print(os.path.splitext(model_path))
```

```
('/content/drive/My Drive/models/employee_turnover_model', '.pkl')
```

```
In [60]: import joblib
model = joblib.load('/content/drive/My Drive/Models/employee_turnover_model.pkl')
test_data = {
    'satisfaction_level': 0.36,
    'last_evaluation': 0.54,
    'number_project': 6,
    'average_monthly_hours': 265,
    'time_spend_company': 5,
    'Work_accident': 0,
    'promotion_last_5years': 1,
```

```
        'sales': 10,  
        'salary': 2  
    }  
    import pandas as pd  
    df = pd.DataFrame([test_data])  
    prediction = model.predict(df)  
    print(prediction)
```

[0]

In []: