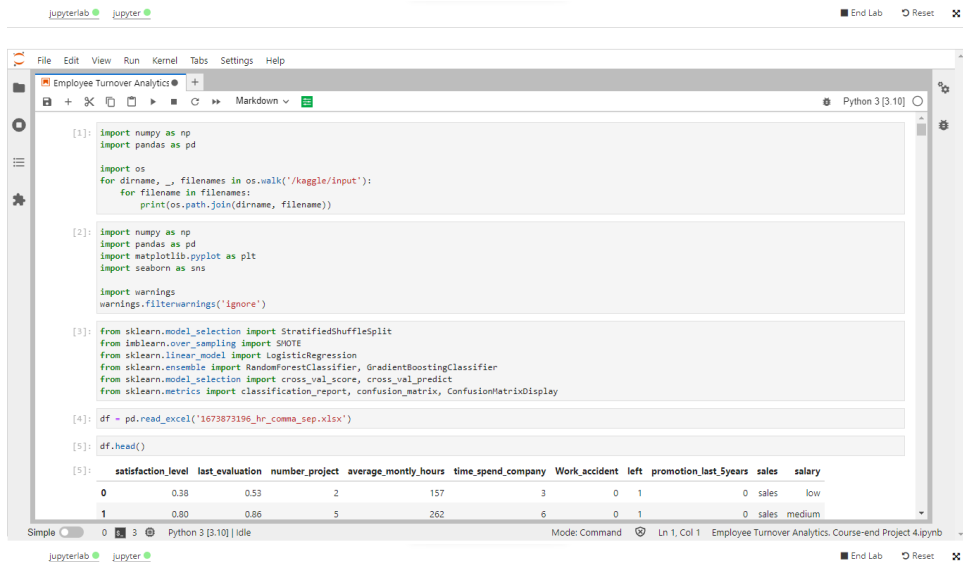


Employee Turnover Analytics.

Course-end Project 4

Screenshots



```
[1]: import numpy as np
import pandas as pd

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

[2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

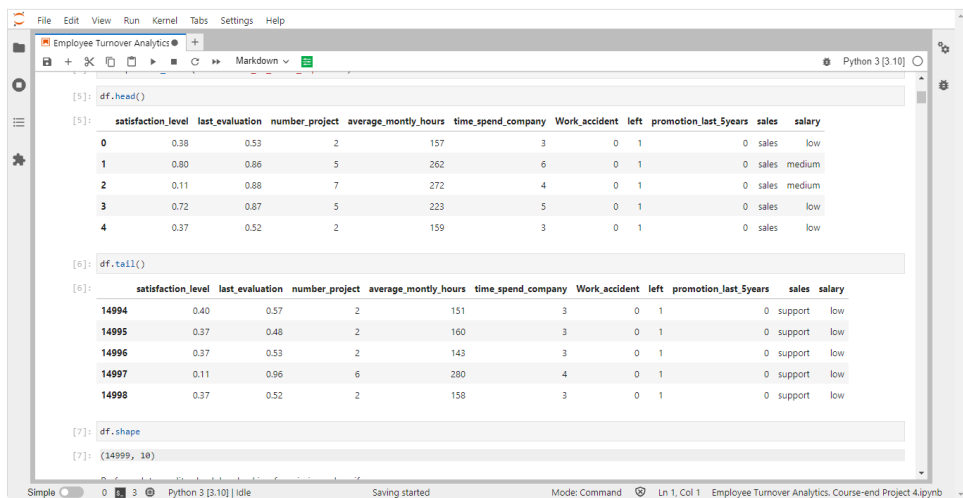
import warnings
warnings.filterwarnings('ignore')

[3]: from sklearn.model_selection import StratifiedShuffleSplit
from imblearn.over_sampling import SMOTE
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.model_selection import cross_val_score, cross_val_predict
from sklearn.metrics import classification_report, ConfusionMatrixDisplay

[4]: df = pd.read_excel('1673873196_hr_comma_sep.xlsx')

[5]: df.head()
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spent_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium



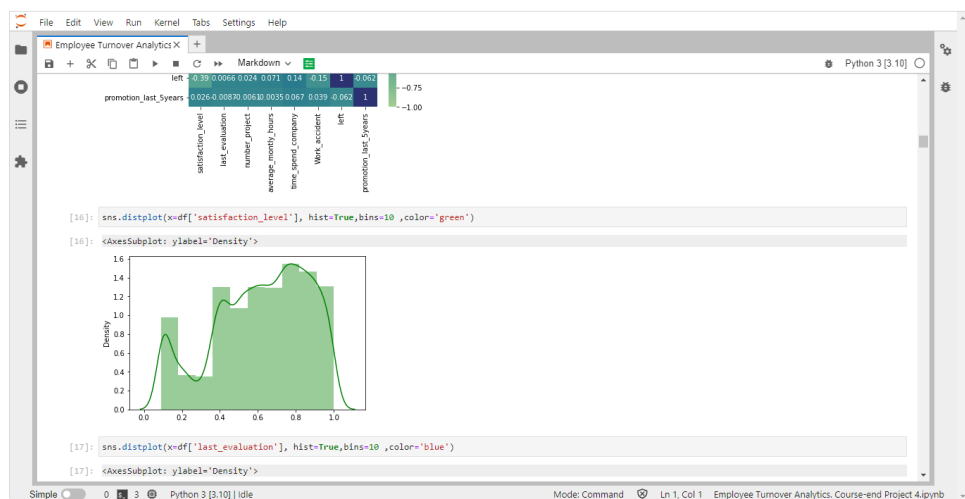
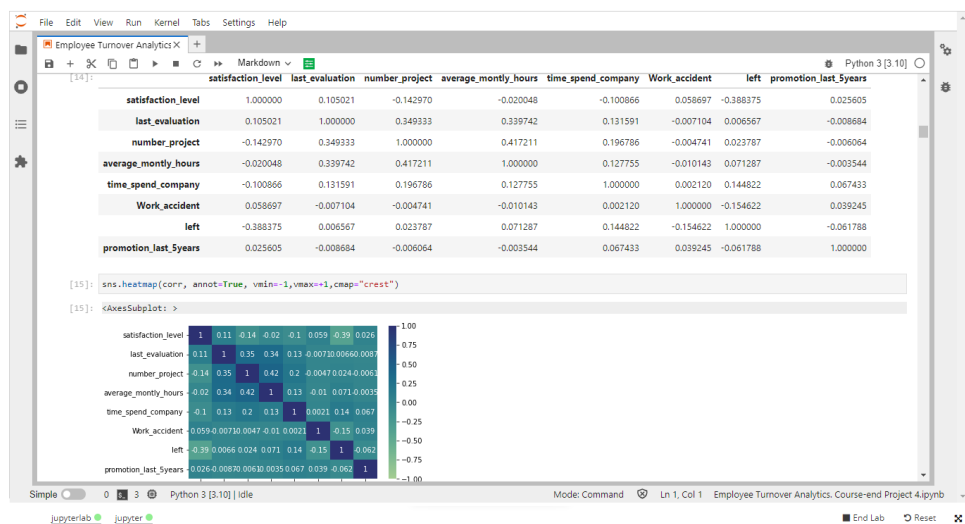
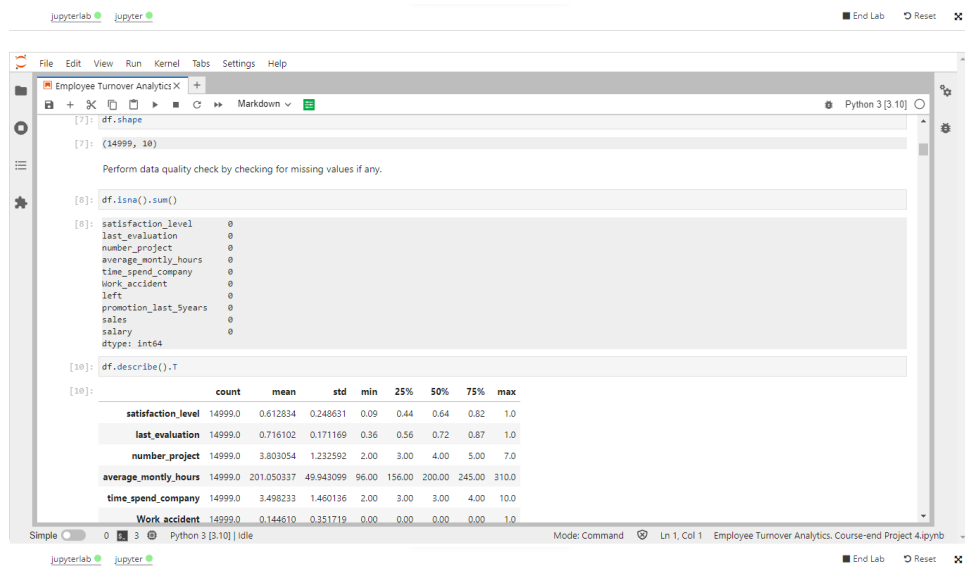
```
[5]: df.head()
```

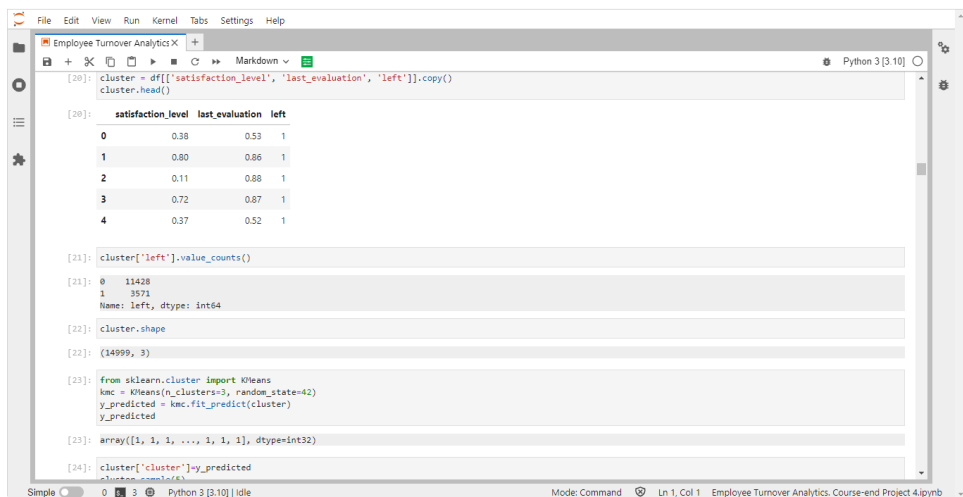
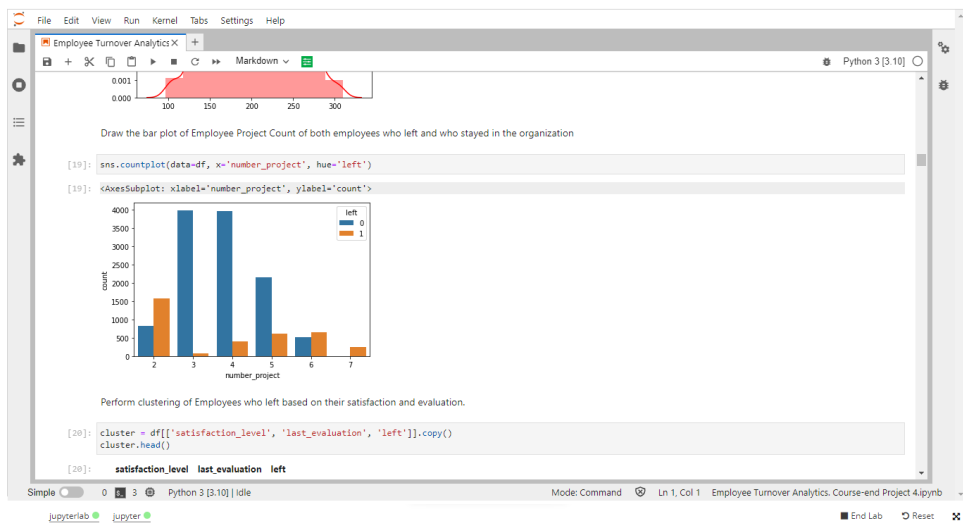
	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spent_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

```
[6]: df.tail()
```

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

```
[7]: df.shape
[7]: (14999, 10)
```





```

jupyterlab jupyter
File Edit View Run Kernel Tabs Settings Help
Employee Turnover Analytics X Python 3 [3.10]
[23]: array([1, 1, ..., 1, 1, 1], dtype=int32)

[24]: cluster['cluster'] = y_predicted
      cluster.sample(5)

[24]:
   satisfaction_level  last_evaluation  left  cluster
11417              0.16              0.46    0      2
14901              0.11              0.93    1      1
1327               0.42              0.47    1      1
14425              0.40              0.53    1      1
11813              0.39              0.91    0      2

[25]: kmc.cluster_centers_

[25]: array([[8.13757657e-01, 7.48231585e-01, 1.62647673e-14],
            [4.40098012e-01, 7.18112574e-01, 1.00000000e+00],
            [4.59096893e-01, 6.88477297e-01, 1.25455202e-14]])

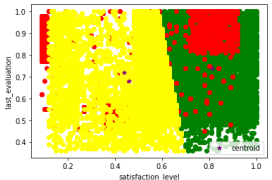
[26]: cluster1 = cluster[cluster.cluster==0]
      cluster2 = cluster[cluster.cluster==1]
      cluster3 = cluster[cluster.cluster==2]
      plt.scatter(cluster1['satisfaction_level'], cluster1['last_evaluation'], color='green')
      plt.scatter(cluster2['satisfaction_level'], cluster2['last_evaluation'], color='red')
      plt.scatter(cluster3['satisfaction_level'], cluster3['last_evaluation'], color='yellow')
      plt.scatter(kmc.cluster_centers_[0], kmc.cluster_centers_[1], color='purple', marker='*', label='centroid')
      plt.xlabel('satisfaction_level')
      plt.ylabel('last_evaluation')
      plt.legend()

Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
jupyterlab jupyter

```

```

jupyterlab jupyter
File Edit View Run Kernel Tabs Settings Help
Employee Turnover Analytics X Python 3 [3.10]
[26]: <matplotlib.legend.Legend at 0x7f902a116f50>



[27]: df.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_spent_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  
Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
jupyterlab jupyter

```

```

jupyterlab jupyter
File Edit View Run Kernel Tabs Settings Help
Employee Turnover Analytics X Python 3 [3.10]
Pre-Process the data by converting categorical columns to numerical columns by Applying get_dummies() to the categorical variables.

[28]: df.head()

[28]:
   satisfaction_level  last_evaluation  number_project  average_monthly_hours  time_spent_company  work_accident  left  promotion_last_5years  department  salary
0              0.38              0.53              2              157              3              0      1              0      sales      low
1              0.80              0.86              5              262              6              0      1              0      sales  medium
2              0.11              0.88              7              272              4              0      1              0      sales  medium
3              0.72              0.87              5              223              5              0      1              0      sales    low
4              0.37              0.52              2              159              3              0      1              0      sales    low

[29]: df.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_spent_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)

Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
jupyterlab jupyter

```

Jupyterlab Jupyter

File Edit View Run Kernel Tabs Settings Help

Employee Turnover Analytics X

Python 3 [3.10]

```
[30]: df['department'].value_counts()

[30]: sales      4140
      technical  2720
      support    2229
      IT         1227
      product_mng  982
      marketing   858
      RandD       787
      accounting  767
      hr          739
      management  630
      Name: department, dtype: int64

[31]: department = pd.get_dummies(df['department'], prefix='department', prefix_sep='_', drop_first=True)
      department

[31]:
```

	department_RandD	department_accounting	department_hr	department_management	department_marketing	department_product_mng	department_sales	department_technical
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	1
...
14994	0	0	0	0	0	0	0	0
14995	0	0	0	0	0	0	0	0
14996	0	0	0	0	0	0	0	0

Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb

Jupyterlab Jupyter

File Edit View Run Kernel Tabs Settings Help

Employee Turnover Analytics X

Python 3 [3.10]

```
[32]: df['salary'].value_counts()

[32]: low      7316
      medium  6446
      high    1237
      Name: salary, dtype: int64

[33]: salary = pd.get_dummies(df['salary'], prefix='salary', prefix_sep='_', drop_first=True)
      salary

[33]:
```

	salary_low	salary_medium	salary_high
0	1	0	0
1	0	1	0
2	0	1	0
3	1	0	0
4	1	0	0
...
14994	1	0	0
14995	1	0	0
14996	1	0	0
14997	1	0	0
14998	1	0	0

14999 rows x 2 columns

Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb

Jupyterlab Jupyter

File Edit View Run Kernel Tabs Settings Help

Employee Turnover Analytics X

Python 3 [3.10]

```
[34]: data = pd.concat([df, department, salary], axis=1)
      data.head()

[34]:
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department	salary	...	depart
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

5 rows x 21 columns

```
[35]: data = data.drop(['department', 'salary'], axis=1)
      data.shape

[35]: (14999, 19)

[36]: data.head()

[36]:
```

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department_RandD	department_accounting	department_hr	department_management	department_marketing	department_product_mng	department_sales	department_technical
0	0.38	0.53	2	157	3	0	1	0	0	0	0	0	0	0	0	
1	0.80	0.86	5	262	6	0	1	0	0	0	0	0	0	0	0	
2	0.11	0.88	7	272	4	0	1	0	0	0	0	0	0	0	0	
3	0.72	0.87	5	223	5	0	1	0	0	0	0	0	0	0	0	

Simple 0 3 Python 3 [3.10] | Idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb

```
jupyterlab jupyter End Lab Reset
```

```
File Edit View Run Kernel Tabs Settings Help
```

```
Employee Turnover Analytics X Python 3 [3.10]
```

```
[37]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 19 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_spent_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_RandD 14999 non-null uint8
 9 department_accounting 14999 non-null uint8
10 department_hr 14999 non-null uint8
11 department_management 14999 non-null uint8
12 department_marketing 14999 non-null uint8
13 department_product_mng 14999 non-null uint8
14 department_sales 14999 non-null uint8
15 department_support 14999 non-null uint8
16 department_technical 14999 non-null uint8
17 salary_low 14999 non-null uint8
18 salary_medium 14999 non-null uint8
dtypes: float64(2), int64(6), uint8(11)
memory usage: 1.1 MB

Do the stratified split of the dataset to train and test.
```

```
[38]: X = data.drop(['left'],axis=1)
      y = data['left']
```

```
Simple 0 3 Python 3 [3.10] idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
```

```
jupyterlab jupyter End Lab Reset
```

```
File Edit View Run Kernel Tabs Settings Help
```

```
Employee Turnover Analytics X Python 3 [3.10]
```

```
[38]: X = data.drop(['left'],axis=1)
      y = data['left']
```

```
[39]: print(X.shape)
      print(y.shape)
      (14999, 18)
      (14999,)
```

```
[40]: sss = StratifiedShuffleSplit(n_splits=5, test_size=0.2, random_state=123)
      sss.get_n_splits(X, y)
```

```
[40]: 5
```

```
[41]: for train,test in sss.split(X,y):
      X_train = X.iloc[train]
      y_train = y.iloc[train]
      X_test = X.iloc[test]
      y_test = y.iloc[test]
      print(y_train.value_counts())
      print(y_test.value_counts())
      0 9142
      1 2857
      Name: left, dtype: int64
      0 2286
      1 714
      Name: left, dtype: int64
```

```
[42]: print(X_train.shape)
      print(y_train.shape)
      print(X_test.shape)
      print(y_test.shape)
```

```
Simple 0 3 Python 3 [3.10] idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
```

```
jupyterlab jupyter End Lab Reset
```

```
File Edit View Run Kernel Tabs Settings Help
```

```
Employee Turnover Analytics X Python 3 [3.10]
```

```
(11999, 18)
(11999,)
(3000, 18)
(3000,)
```

Upsample the train dataset using SMOTE technique from the imblearn module. que.

```
[43]: smote = SMOTE(random_state = 11)
      X_train, y_train = smote.fit_resample(X_train, y_train)
```

```
[44]: print(X_train.shape)
      print(y_train.shape)
      (18284, 18)
      (18284,)
```

Train a Logistic Regression model and apply a 5-Fold CV and plot the classification report.

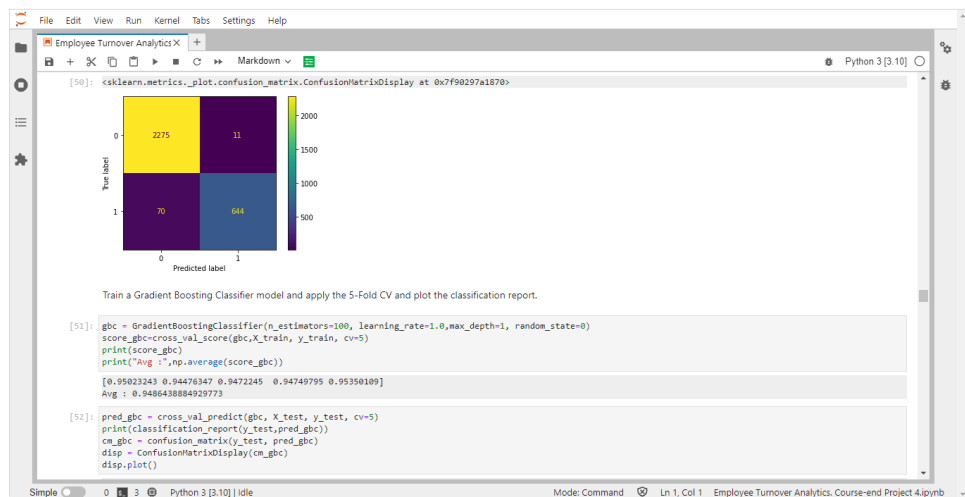
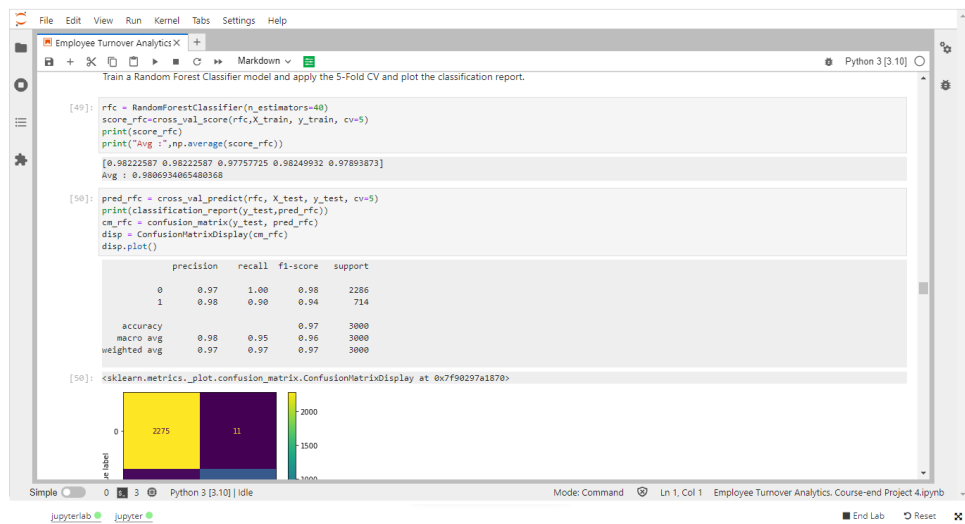
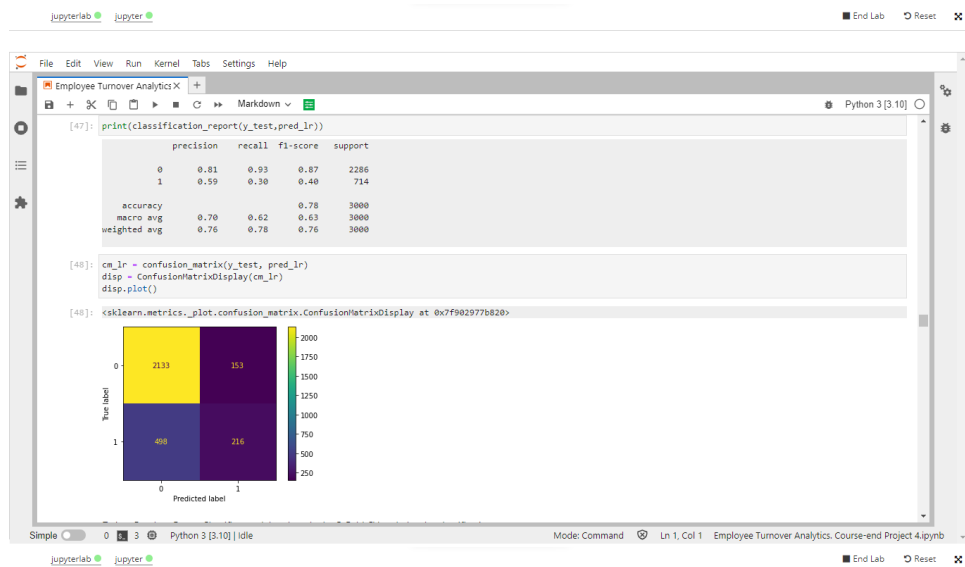
```
[45]: lr = LogisticRegression(solver='liblinear',multi_class='ovr')
      score_lr=cross_val_score(lr,X_train, y_train, cv=5)
      print(score_lr)
      print("Avg :",np.average(score_lr))
      [0.74651354 0.77796008 0.81104731 0.81596937 0.80661926]
      Avg : 0.7916219097214119
```

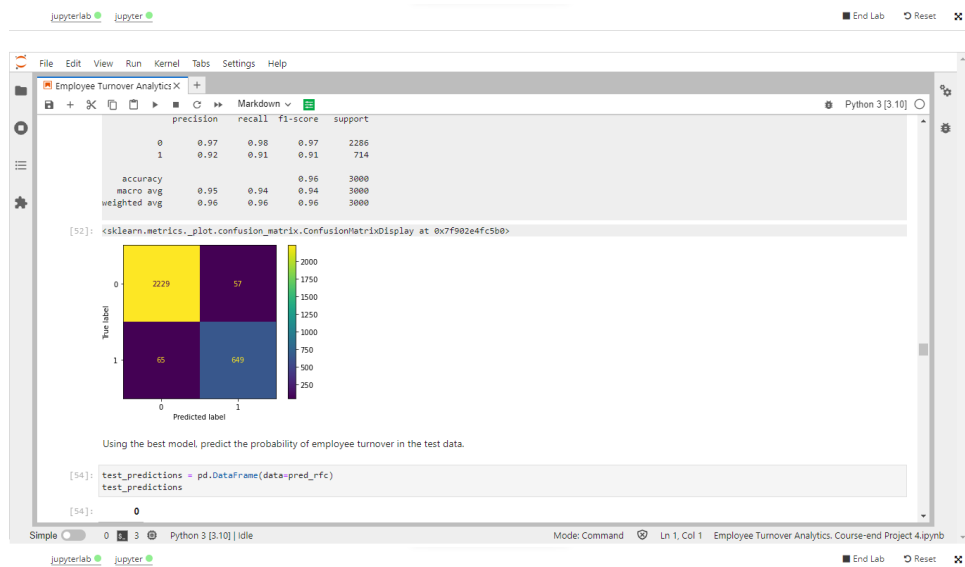
```
[46]: pred_lr = cross_val_predict(lr, X_test, y_test, cv=5)
      pred_lr
```

```
[46]: array([0, 0, 0, ..., 0, 1, 0])
```

```
[47]: print(classification_report(y_test,pred_lr))
```

```
Simple 0 3 Python 3 [3.10] idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb
```





Employee Turnover Analytics X

[54]: test_predictions = pd.DataFrame(data=pred_rfc)
test_predictions

[54]:

```
0
0 1
1 1
2 0
3 0
4 0
...
2995 0
2996 1
2997 0
2998 0
2999 1
```

3000 rows x 1 columns

[55]: test_predictions.rename(columns={0:'predictions'},inplace=True)
test_predictions.head()

[55]:

```
predictions
0      1
1      1
2      0
3      0
4      0
```

Simple 0 3 Python 3 [3.10] idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb

Employee Turnover Analytics X

[55]: test_predictions.rename(columns={0:'predictions'},inplace=True)
test_predictions.head()

[55]:

```
predictions
0      1
1      1
2      0
3      0
4      0
```

[56]: prob = cross_val_predict(rfc, X_test, y_test, cv=5, method='predict_proba')
keep probabilities for the positive outcome only
prob = prob[:, 1]
prob

[56]: array([1. , 0.95 , 0.025, ..., 0.05 , 0.05 , 0.95])

[57]: probability = pd.DataFrame(data=prob)
probability.head()

[57]:

```
0
0 1.000
1 0.950
2 0.025
3 0.000
```

Simple 0 3 Python 3 [3.10] idle Mode: Command Ln 1, Col 1 Employee Turnover Analytics. Course-end Project 4.ipynb


```
probability.rename(columns={0:'probability'},inplace=True)
probability.head()

probability
0    1.000
1    0.950
2    0.025
3    0.000
4    0.000

[59]: len(probability)
[59]: 3000

Based on the below probability score range, categorize the employees into four zones and suggest your thoughts on the retention strategies for each zone.
■ Safe Zone (Green) (Score < 20%)
■ Low Risk Zone (Yellow) (20% < Score < 60%)
■ Medium Risk Zone (Orange) (60% < Score < 90%)
■ High Risk Zone (Red) (Score > 90%).

[60]: # create a list of our conditions
conditions = [
    (probability['probability'] <= 0.2),
    (probability['probability'] > 0.2) & (probability['probability'] <= 0.6),
    (probability['probability'] > 0.6) & (probability['probability'] <= 0.9),
    (probability['probability'] > 0.9)
]
```

```
# create a list of the values we want to assign for each condition
values = ['Safe Zone (Green)', 'Low Risk Zone (Yellow)', 'Medium Risk Zone (Orange)', 'High Risk Zone (Red)']

# create a new column and use np.select to assign values to it using our lists as arguments
probability['zone'] = np.select(conditions, values)

# display updated DataFrame
probability.head()

[60]: probability    zone
0    1.000  High Risk Zone (Red)
1    0.950  High Risk Zone (Red)
2    0.025  Safe Zone (Green)
3    0.000  Safe Zone (Green)
4    0.000  Safe Zone (Green)

[61]: print(X_test.shape)
print(test_predictions.shape)
print(probability.shape)
(3000, 18)
(3000, 1)
(3000, 2)

[62]: X_test = X_test.reset_index()

[63]: new_test_df = pd.concat([X_test, test_predictions, probability], axis=1)
```

```
[63]: new_test_df = pd.concat([X_test, test_predictions, probability], axis=1)
new_test_df.head()

[63]: index  satisfaction_level  last_evaluation  number_project  average_monthly_hours  time_spend_company  Work_accident  promotion_last_5years  department_RandD  departmen

0    439                0.41              0.52              2                136                3                0                0                0

1    649                0.46              0.50              2                156                3                0                0                0

2    8478               0.58              0.63              5                191                3                1                0                0

3    13225              0.52              0.89              3                188                6                0                0                0

4    7962               0.74              0.54              4                167                2                0                0                0

5 rows x 22 columns

[64]: new_test_df['zone'].value_counts()

[64]: Safe Zone (Green)    2249
```

