Data Collection and Preprocessing Phase

Date	13 JULY 2024
Team ID	SWTID1720174957
Project Title	Human resource management employee promortion prediction using machine learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
Data Overview	Basic statistics, dimensions, and structure of the data.
Univariate Analysis	Exploration of individual variables (mean, median, mode, etc.).
Bivariate Analysis	Relationships between two variables (correlation, scatter plots).
Multivariate Analysis	Patterns and relationships involving multiple variables.
Outliers and Anomalies	Identification and treatment of outliers.
Data Preprocessing Code Screenshots	

Loading Data	`df = pd.read_csv('C:\Dataset\emp_promotion.csv')`
	`print('Shape of train data {}'.format(df.shape))`
	`Shape of train data (54808, 14)`

Handling Missing Data

```
df.isnull().sum()
```

```
department                0
education              2409
no_of_trainings           0
age                       0
previous_year_rating   4124
length_of_service         0
KPIs_met >80%             0
awards_won?               0
avg_training_score        0
is_promoted               0
dtype: int64
```

```
print(df['education'].value_counts())
df['education']=df['education'].fillna(df['education'].mode()[0])
```

```
education
Bachelor's          36669
Master's & above    14925
Below Secondary       805
Name: count, dtype: int64
```

```
#Replacing nan with mode

print(df['previous_year_rating'].value_counts())
df['previous_year_rating']=df['previous_year_rating'].fillna(df['previous_year_rating'].mode()[0])
```

```
previous_year_rating
3.0    18618
5.0    11741
4.0     9877
1.0     6223
2.0     4225
Name: count, dtype: int64
```

Data Transformation

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Select numerical columns
numerical_columns = ['no_of_trainings', 'age', 'previous_year_rating',
                     'length_of_service', 'KPIs_met >80%', 'awards_won?',
                     'avg_training_score']

# Standard scaling
scaler = StandardScaler()
data_standard_scaled = pd.read_csv('C:\Dataset\emp_promotion.csv')
data_standard_scaled[numerical_columns] = scaler.fit_transform(data_standard_scaled[numerical_columns])

# Min-Max normalization
min_max_scaler = MinMaxScaler()
data_min_max_scaled = pd.read_csv('C:\Dataset\emp_promotion.csv')
data_min_max_scaled[numerical_columns] = min_max_scaler.fit_transform(data_min_max_scaled[numerical_columns])

# Display the first few rows of the scaled and normalized data
print("Standard Scaled Data:\n", data_standard_scaled.head())
print("\nMin-Max Normalized Data:\n", data_min_max_scaled.head())
```

```
Standard Scaled Data:
    employee_id         department     region         education gender  \
0        65438  Sales & Marketing   region_7  Master's & above      f
1        65141         Operations  region_22        Bachelor's      m
2         7513  Sales & Marketing  region_19        Bachelor's      m
3         2542  Sales & Marketing  region_23        Bachelor's      m
4        48945         Technology  region_26        Bachelor's      m

  recruitment_channel  no_of_trainings       age  previous_year_rating  \
0            sourcing        -0.415276  0.025598              1.326008
1               other        -0.415276 -0.627135              1.326008
2            sourcing        -0.415276 -0.104948             -0.261318
3               other         1.226063  0.547785             -1.848645
4               other        -0.415276  1.331064             -0.261318

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0           0.500460       1.356878    -0.154018           -1.075931
1          -0.437395      -0.736986    -0.154018           -0.253282
2           0.265996      -0.736986    -0.154018           -1.001145
3           0.969387      -0.736986    -0.154018           -1.001145
4          -0.906322      -0.736986    -0.154018            0.718939

   is_promoted
0            0
1            0
2            0
3            0
4            0
```

```
Min-Max Normalized Data:
    employee_id         department     region         education gender  \
0         65438  Sales & Marketing   region_7  Master's & above      f
1         65141         Operations  region_22        Bachelor's      m
2          7513  Sales & Marketing  region_19        Bachelor's      m
3          2542  Sales & Marketing  region_23        Bachelor's      m
4         48945         Technology  region_26        Bachelor's      m

  recruitment_channel  no_of_trainings    age  previous_year_rating  \
0            sourcing         0.000000  0.375                   1.0
1               other         0.000000  0.250                   1.0
2            sourcing         0.000000  0.350                   0.5
3               other         0.111111  0.475                   0.0
4               other         0.000000  0.625                   0.5

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0           0.194444            1.0          0.0            0.166667
1           0.083333            0.0          0.0            0.350000
2           0.166667            0.0          0.0            0.183333
3           0.250000            0.0          0.0            0.183333
4           0.027778            0.0          0.0            0.566667

   is_promoted
0            0
1            0
2            0
3            0
4            0
```

Feature Engineering

```
import pandas as pd

# Load the CSV file

data = pd.read_csv('C:\Dataset\emp_promotion.csv')

# Create a new feature 'is_high_performer'
data['is_high_performer'] = data.apply(lambda row: 1 if row['KPIs_met >80%'] == 1 and row['awards_won?'] == 1 else 0, axis=1)

# Create an 'age_group' feature
bins = [20, 30, 40, 50, 60]
labels = ['20-29', '30-39', '40-49', '50-59']
data['age_group'] = pd.cut(data['age'], bins=bins, labels=labels, right=False)

# Modify the 'education' feature to fewer categories
data['education'] = data['education'].replace({
    "Master's & above": "Postgraduate",
    "Bachelor's": "Undergraduate",
    "Below Secondary": "Secondary"
})

# Display the first few rows to verify changes
print(data.head())
```

```
   employee_id         department     region      education gender  \
0        65438  Sales & Marketing   region_7   Postgraduate      f
1        65141         Operations  region_22  Undergraduate      m
2         7513  Sales & Marketing  region_19  Undergraduate      m
3         2542  Sales & Marketing  region_23  Undergraduate      m
4        48945         Technology  region_26  Undergraduate      m

  recruitment_channel  no_of_trainings  age  previous_year_rating  \
0            sourcing                1   35                   5.0
1               other                1   30                   5.0
2            sourcing                1   34                   3.0
3               other                2   39                   1.0
4               other                1   45                   3.0

   length_of_service  KPIs_met >80%  awards_won?  avg_training_score  \
0                  8              1            0                  49
1                  4              0            0                  60
2                  7              0            0                  50
3                 10              0            0                  50
4                  2              0            0                  73

   is_promoted  is_high_performer age_group
0            0                  0     30-39
1            0                  0     30-39
2            0                  0     30-39
3            0                  0     30-39
4            0                  0     40-49
```

Save Processed Data

```
import pickle
pickle.dump(rf, open('promotion_model.pkl', 'wb'))
```

	Today promotion_model.pkl 14-07-2024 17:33 PKL File 1,11,024 KB final 14-07-2024 17:35 Jupyter Source File 2,287 KB emp_promotion 14-07-2024 16:50 Microsoft Excel Co... 3,672 KB