```
# -*- coding: utf-8 -*- """Data Analysis Question of HRAnalytics.ipynb Automatically generated by Colaboratory. Original file is
located at https://colab.research.google.com/drive/1_p7IRgckM4jZrnSLSa-UAB-QEoawYrpE #Project HR Analytics Project.#
#Describe:# This case study aims to model the probability of attrition of each employee from the HR Analytics Dataset, available on
Kaggle. Its conclusions will allow the management to understand which factors urge the employees to leave the company and
which changes should be made to avoid their departure. All the files of this project are saved in a GitHub repository. link:
https://github.com/sameerCoder/DATA_ANALYST_DATASETS/tree/main/HrAnalytics """ from google.colab import drive
#drive.mount('/content/drive') """**Importing Libraries and Dataset**""" import numpy as np import pandas as pd import
matplotlib.pyplot as plt import seaborn as sns # reading the data train =
pd.read_csv('https://raw.githubusercontent.com/sameerCoder/DATA_ANALYST_DATASETS/main/HrAnalytics/HrAnalytics_train.csv')
test =
pd.read_csv('https://raw.githubusercontent.com/sameerCoder/DATA_ANALYST_DATASETS/main/HrAnalytics/HrAnalytics_test.csv')
# getting their shapes #____________fill blank with code_____________________________ """**Data Analysis**""" # print head of
train and test df #____________fill blank with code_____________________________ # describing the training set . all columns
should display in result. #____________fill blank with code_____________________________ #print info of train and test
#____________fill blank with code_____________________________ # checking if there is any NULL value in the dataset
#____________fill blank with code_____________________________ #____________fill blank with
code_____________________________ """**UNi-variate Data Visualization**""" # looking at the most popular departments with tittle
Most Popular Departments # use and import matplotlib, wordcloud & stopwords from wordcloud import WordCloud from wordcloud
import STOPWORDS # checkig the no. of Employees Promoted #____________fill blank with
code_____________________________ # finding the %age of people promoted #____________fill blank with
code_____________________________ #plotting a scatter plot #title - plot to show the gap in Promoted and Non-Promoted
Employees #use train['is_promoted'] #____________fill blank with code_____________________________ # checking the
distribution of the avg_training score of the Employees # title - Distribution of Training Score among the Employees #use
train['avg_training_score'] #____________fill blank with code_____________________________ # plotting a donut chart for
visualizing each of the recruitment channel's share # title 'Showing a Percentage of employees who won awards' # use plt.Circle #
labels = "Awards Won", "NO Awards Won" # add legend (Awards Won and NO Awards Won) #____________fill blank with
code_____________________________ # find the counts whose 'KPIs_met >80%' #____________fill blank with
code_____________________________ # plotting a pie chart # labels = "Not Met KPI > 80%", "Met KPI > 80%" # title 'A Pie Chart
Representing Gap in Employees in terms of KPI' # display legend #____________fill blank with
code_____________________________ # checking the distribution of length of service # title 'Distribution of length of service
among the Employees' #____________fill blank with code_____________________________ # 'Distribution of Previous year rating
of the Employees' #____________fill blank with code_____________________________ # checking the distribution of age of
Employees in the company #____________fill blank with code_____________________________ # checking the different no. of
training done by the employees # use Violinplot for the train['no_of_trainings'] column # title 'No. of trainings done by the
Employees' #____________fill blank with code_____________________________ # checking the different types of recruitment
channels for the company # use value_counts() #____________fill blank with code_____________________________ # plotting a
donut chart for visualizing each of the recruitment channel's share # use plt.Circle and plt.pie #labels = "Others", "Sourcing",
"Reffered" #____________fill blank with code_____________________________ # checking the most popular education degree
among the employees # title 'Most Popular Degrees among the Employees' from wordcloud import WordCloud from wordcloud
import STOPWORDS # checking the gender gap # count male and female #____________fill blank with
code_____________________________ # plotting a pie chart # title A Pie Chart Representing GenderGap # legend Male & Female
#____________fill blank with code_____________________________ # checking the different regions of the company # title
'Different Regions in the company' #____________fill blank with code_____________________________ """**Bi-varaiate Data
Visualization**""" # scatter plot between average training score and is_promoted # use crosstab in two columns
train['avg_training_score'], train['is_promoted'] #____________fill blank with code_____________________________ """**As, the
Training Scores Increases, the chances of Promotion Increases Highly**""" # checking dependency of different regions in promotion
# use pd.crosstab train['region'], train['is_promoted'] # title 'Dependency of Regions in determining Promotion of Employees'
#____________fill blank with code_____________________________ """**The above graph shows that there is no biasedness over
regions in terms of Promotion as all the regions share promotions almost equally.**""" # dependency of awards won on promotion #
pd.crosstab train['awards_won?'], train['is_promoted'] #____________fill blank with code_____________________________
"""**There is a very good chance of getting promoted if the employee has won an award**""" #dependency of KPIs with Promotion
data = pd.crosstab(train['KPIs_met >80%'], train['is_promoted']) data.div(data.sum(1).astype('float'), axis = 0).plot(kind = 'bar',
stacked = True, figsize = (10, 8), color = ['pink', 'darkred']) plt.title('Dependency of KPIs in determining Promotion', fontsize = 30)
plt.xlabel('KPIs Met or Not', fontsize = 20) plt.legend() plt.show() #____________fill blank with
code_____________________________ """**Again Having a good KPI score increases the chances of getting promoted in the
company.**""" # checking dependency on previous years' ratings # pd.crosstab(train['previous_year_rating'], train['is_promoted'])
#____________fill blank with code_____________________________ """**The Above Graph clearly suggests that previous ratings
matter a lot, if the ratings are high, the chances of being promoted in the company increases and there is completely no promotion
for the employees with previous year ratings = 0**""" # checking how length of service determines the promotion of employees
#data = pd.crosstab(train['length_of_service'], train['is_promoted']) # checking dependency of age factor in promotion of employees
"""**This is Very Impressive that the company promotes employees of all the ages equally even the freshers have equal share of
promotion and also the senior citizen employees are getting the equal share of Promotion in the Company**""" # checking which
department got most number of promotions #data = pd.crosstab(train['department'], train['is_promoted'])
#data.div(data.sum(1).astype('float'), axis = 0).plot(kind = 'bar', stacked = True, figsize = (20, 8), color = ['orange', 'lightgreen'])
"""**Again, Each of the departments have equal no. of promotions showing an equal developement in each of the departments of
the company.**""" # checking dependency of gender over promotion """**The above plot shows that there is no partiality between
males and females in terms of promotion** **Data Pre-processing** """ # filling missing values # again checking if there is any Null
value left in the data # filling missing values # again checking if there is any Null value left in the data # removing the employee_id
column # saving the employee_id emp_id = test['employee_id'] # removing the employee_id column # print all the columns #
defining the test set x_test = test # one hot encoding for the test set # use pd.get_dummies in x_test # print all the columns #
splitting the train set into dependent and independent sets x = # all rows and last columns in train dataframe y = # all rows and last
columns in train dataframe # print the shape of x & y # Do one hot encoding for the train set # pd.get_dummies(x) #print all columns
# Thank you!!!
```