Module 6: Data Manipulation

Case Study Solution

edureka!

Case Study Solution

Domain – HR

focus – Insights from data

Business challenge/requirement
SFO Public Department - referred to as SFO has captured all the salary data of its employees from year 2011-2014. Now we are in year 2015 and the organization is facing some financial crisis. As first step HR wants to rationalize employee cost to save payroll budget. You have to do data manipulation and analysis on the salary data to answer specific questions for cost savings.

Key issues
Cost can be saved by figuring out the key pockets of high salaries

Considerations
NONE

Data volume
- Approx 150K records across files

Additional information
- NA

Business benefits
Save at least 10% of employee cost by identifying and letting them go

Approach to Solve
You have to use fundamentals of Pandas covered in module 6 and answer following 5 Questions

1. Compute how much total salary cost has increased from year 2011 to 2014
2. Which Job Title in Year 2014 has highest mean salary?
3. How much money could have been saved in Year 2014 by stopping OverTimePay?

4. Which are the top 5 common job in Year 2014 and how much do they cost SFO ?
5. Who was the top earning employee across all the years?

Enhancements for code

You can try these enhancements in code

1. Which are the last 5 common job in Year 2014 and how much do they cost SFO?
2. In year 2014 OverTimePay was what percentage of TotalPayBenefits
3. Which Job Title in Year 2014 has lowest mean salary?

Solution

```
import pandas as pd

# Set the option on how to display float
# To See the impact of this command comment it out and check outputs
pd.set_option('display.float_format', lambda x: '%.3f' % x)

#  Read Salaries.csv as a dataframe called salary
salary = pd.read_csv('Salaries.csv')

# Check the basic data

print(salary.head())

# Get more info on data & Verify that 148,648 records are there

print (salary.info())


# Get Some statistical summary of data
print (salary.describe())


# Check the total salary cost per year and see how it has increased over years

sum_year = salary.groupby('Year').sum()['TotalPayBenefits']
print ( sum_year)
```

```
# Question 1: Compute how much total salary cost has increased from year 2011 to
2014

# Check the mean salary cost per year and see how it has increased per year

mean_year = salary.groupby('Year').mean()['TotalPayBenefits']
print ( mean_year)

# To Effectively Reduce Cost Check which is the highest mean paying job Title Wise
# Question 2: Which Job Title in Year 2014 has highest mean salary
mean_title_year = salary.groupby(['Year','JobTitle']).mean()['TotalPayBenefits']
print ( mean_title_year)


# Question 3: How much money could have been saved in Year 2014 by stopping
OverTimePay

over_time_pay_year = salary.groupby('Year').sum()['OvertimePay']
print ( over_time_pay_year)


# Question 4: Which are the top 5 common job in Year 2014 and how much do they cost
SFO -- Little Bit Tricky
top_job_title = salary[salary['Year'] == 2014]['JobTitle'].value_counts().head(5)
print (top_job_title)

# Uncomment this and check what it prints
#print (type(top_job_title))

# Calculate the Cost
sum_cost = 0
for index,value in top_job_title.iteritems():
   print(index,value)
   sum_cost += sum(salary[ (salary['Year']== 2014) & (salary['JobTitle'] ==
index)]['TotalPayBenefits'])

print (" Total Cost of Top 5 Jobs in Year 2014 ", sum_cost)

# Question 5: Who was the top earning employee across all the years
# This is compute intensive -- might take some time to execute

top_sal = salary.groupby('EmployeeName').sum()['TotalPayBenefits']
print((top_sal.sort_values(axis=0)))

# You can improve the code above
```

```
print ("End of Cast Study............")
```