Assignment Code: DA-AG-007

Foundations of Machine Learning and EDA| **Assignment**

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 200

Question 1 : What is the difference between AI, ML, DL, and Data Science? Provide a brief explanation of each.

(Hint: Compare their scope, techniques, and applications for each.) **Answer:**

1. Artificial Intelligence (AI) is the ultimate goal of making machines intelligent enough to simulate human thinking and behavior.

2. Machine Learning (ML) is a method of achieving AI, where a system learns directly from data without being explicitly programmed.

3. Deep Learning (DL) is a sub-field of ML that uses large, multi-layered artificial neural networks to analyze complex data patterns automatically.

4. Data Science (DS) is an umbrella discipline that uses statistics, data analysis, and ML/DL techniques to extract valuable insights and make data-driven decisions.

Question 2: Explain overfitting and underfitting in ML. How can you detect and prevent them?

Hint: Discuss bias-variance tradeoff, cross-validation, and regularization techniques.

Answer:

- Overfitting happens when a model learns the training data too much, including noise and mistakes.
- It performs very well on training data but performs poorly on new test data.
- Underfitting happens when a model is too simple and cannot learn the patterns properly.
- It performs poorly on both training data and test data.
- Bias-variance tradeoff means balancing between too simple (high bias) and too complex (high variance) models.
- To detect overfitting or underfitting, compare training accuracy and test accuracy using cross-validation.
- If training accuracy is high but test accuracy is low, it is overfitting.
- If both accuracies are low, it is underfitting.
- To prevent overfitting, use techniques like regularization, dropout, pruning, and more data.
- To prevent underfitting, use a more complex model or train for more time.

Question 3:How would you handle missing values in a dataset? Explain at least three methods with examples.

Hint: Consider deletion, mean/median imputation, and predictive modeling.

Answer:

- Missing values mean some data entries are empty or not recorded.
- One method is deletion, where rows with missing values are removed.
- For example, if a row has no age value, we delete that row.
- This is useful when missing data is very small.
- Another method is mean or median imputation.
- We replace the missing number with the average (mean) or middle value (median) of that column.

- For example, if some people's height is missing, we put the average height in place of the missing ones.
- A third method is predictive modeling.
- We use machine learning models to predict the missing values.

Question 4:What is an imbalanced dataset? Describe two techniques to handle it (theoretical + practical).

Hint: Discuss SMOTE, Random Under/Oversampling, and class weights in models.

Answer:

- An imbalanced dataset means one class has many more samples than the other classes.
- For example, in fraud detection, normal transactions are many, but fraud cases are very few.
- This causes the model to learn more about the majority class and ignore the minority class.
- One technique is Random Oversampling.
- It increases the number of minority class samples by duplicating them.
- For example, if we have only 50 fraud cases, we duplicate them to make 200 cases.
- The benefit is better learning of minority class patterns.
- Another technique is SMOTE (Synthetic Minority Over-sampling Technique).
- Instead of copying, it creates new synthetic minority samples based on existing ones.
- For example, it generates new fraud samples using similar fraud behaviors.
- This helps improve model performance without exact duplicates.
- Models can also use class weights.
- We tell the algorithm to give more importance or penalty to wrong predictions of minority class.

Question 5: Why is feature scaling important in ML? Compare Min-Max scaling and Standardization.

Hint: Explain impact on distance-based algorithms (e.g., KNN, SVM) and gradient descent.

Answer:

- Feature scaling means adjusting the range of data values to be similar.
- It is important because some algorithms like KNN, SVM, and gradient descent work better when features have similar scales.
- Without scaling, features with larger values dominate the learning process.
- Min-Max Scaling converts values to a fixed range, usually 0 to 1.
- Example: height values from 150 to 190 become 0 to 1 range.
- It keeps the shape of the original distribution but compresses the values.
- Standardization converts data to have mean 0 and standard deviation 1.
- It centers the data around zero and helps when data has outliers.
- It is useful for algorithms using distance or gradient-based optimization.
- Both methods help models learn faster and perform better.

Question 6: Compare Label Encoding and One-Hot Encoding. When would you prefer one over the other?

Hint: Consider categorical variables with ordinal vs. nominal relationships.

Answer:

- Label Encoding converts each category into a number like 0, 1, 2, 3.
- It is useful when categories have an order (ordinal data).
- Example: Small = 0, Medium = 1, Large = 2.
- The model understands size ranking using these numbers.
- One-Hot Encoding creates separate binary columns for each category.
- Example: Red = [1,0,0], Blue = [0,1,0], Green = [0,0,1].
- It is useful when categories have no order (nominal data).
- It avoids giving any ranking meaning to categories.
- Prefer Label Encoding for ordered data to keep the natural ranking.
- Prefer One-Hot Encoding for unordered data to prevent the model from misunderstanding category importance.

Question 7: Google Play Store Dataset

a). Analyze the relationship between app categories and ratings. Which categories have the highest/lowest average ratings, and what could be the possible reasons?
Dataset: https://github.com/MasteriNeuron/datasets.git **(Include your Python code and output in the code box below.)**

Answer:

```
   import pandas as pd
import numpy as np


url = 'https://raw.githubusercontent.com/MasteriNeuron/datasets/main/googleplaystore.csv'
df = pd.read_csv(url)

print("Rows, Columns:", df.shape)
print(df.columns.tolist())
print(df.head(3))

df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')
df = df.dropna(subset=['Rating'])
```

```
df = df[(df['Rating'] >= 0) & (df['Rating'] <= 5)]

cat_stats = df.groupby('Category')['Rating'].agg(['count','mean','median','std']).reset_index()
cat_stats = cat_stats.sort_values('mean', ascending=False)

MIN_COUNT = 30
cat_stats_filtered = cat_stats[cat_stats['count'] >= MIN_COUNT].sort_values('mean', ascending=False)

print("\nTop 5 categories by average rating:")
print(cat_stats_filtered.head(5).to_string(index=False))

print("\nBottom 5 categories by average rating:")
print(cat_stats_filtered.tail(5).to_string(index=False))

cat_stats_filtered.to_csv('category_rating_stats.csv', index=False)

#OUTPUT:

Rows, Columns: (10841, 13)
['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs', 'Type',
'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current Ver',
'Android Ver']
                                                 App        Category
Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN
4.1
1                                Coloring book moana  ART_AND_DESIGN
3.9
2  U Launcher Lite – FREE Live Cool Themes, Hide ...  ART_AND_DESIGN
4.7

  Reviews  Size    Installs  Type Price Content Rating  \
0     159   19M     10,000+  Free     0       Everyone
1     967   14M    500,000+  Free     0       Everyone
2   87510  8.7M  5,000,000+  Free     0       Everyone

                     Genres      Last Updated Current Ver   Android Ver
0               Art & Design   January 7, 2018       1.0.0  4.0.3 and up
1  Art & Design;Pretend Play  January 15, 2018       2.0.0  4.0.3 and up
2               Art & Design    August 1, 2018       1.2.4  4.0.3 and up

Top 5 categories by average rating:
           Category  count     mean  median      std
             EVENTS     45 4.435556     4.5 0.419499
          EDUCATION    155 4.389032     4.4 0.251894
     ART_AND_DESIGN     62 4.358065     4.4 0.358297
BOOKS_AND_REFERENCE    178 4.346067     4.5 0.429046
    PERSONALIZATION    314 4.335987     4.4 0.352732

Bottom 5 categories by average rating:
```

```
           Category  count     mean  median      std
          LIFESTYLE    314 4.094904     4.2 0.693907
      VIDEO_PLAYERS    160 4.063750     4.2 0.551098
MAPS_AND_NAVIGATION    124 4.051613     4.2 0.519926
              TOOLS    734 4.047411     4.2 0.616143
             DATING    195 3.970769     4.1 0.630510
```

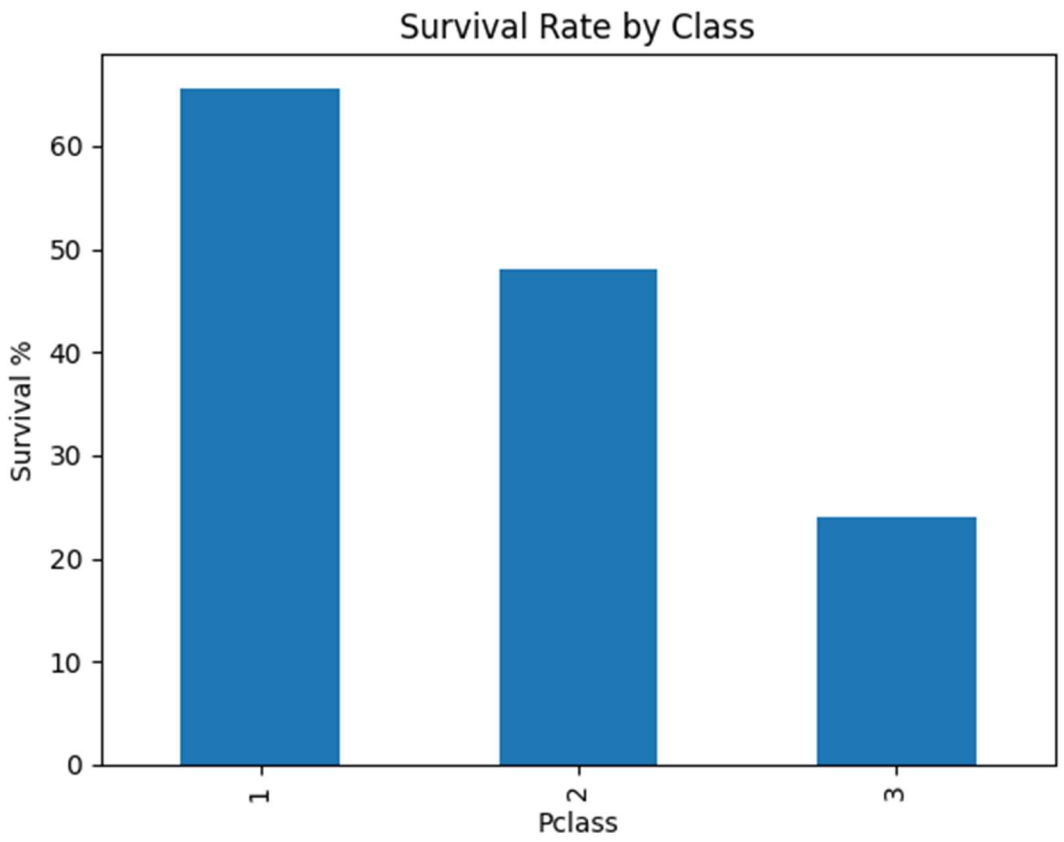
Question 8: **Titanic Dataset**

a) Compare the survival rates based on passenger class (Pclass). Which class had the highest survival rate, and why do you think that happened?

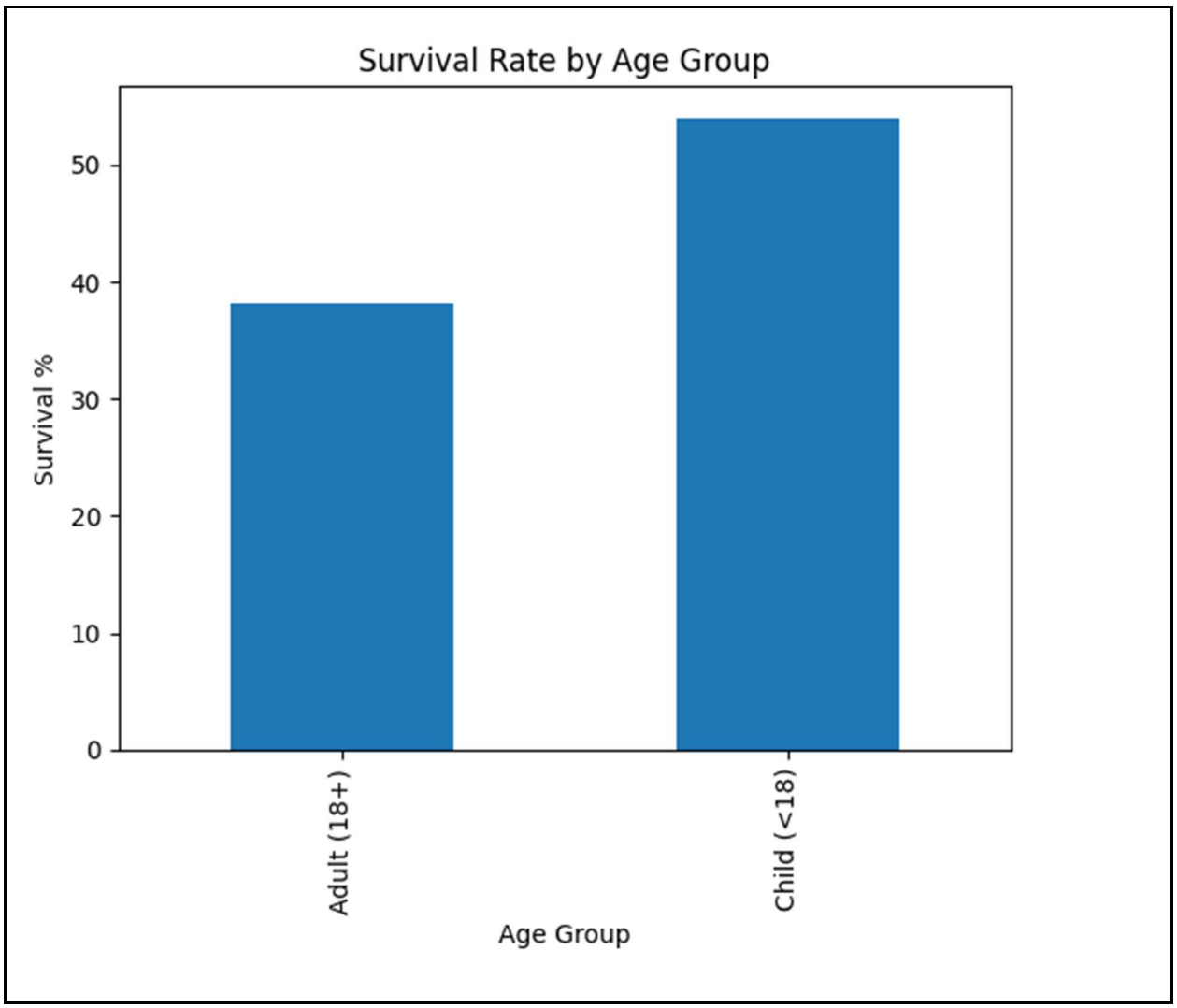
b) Analyze how age (Age) affected survival. Group passengers into children (Age < 18) and adults (Age ≥ 18). Did children have a better chance of survival? Dataset: https://github.com/MasteriNeuron/datasets.git **(Include your Python code and output in the code box below.) Answer:**

```
 url = "https://raw.githubusercontent.com/pandas-
dev/pandas/master/doc/data/titanic.csv"

import pandas as pd
import matplotlib.pyplot as plts

url = "https://raw.githubusercontent.com/pandas-
dev/pandas/master/doc/data/titanic.csv"

df = pd.read_csv(url)
print("Columns:", df.columns.tolist())
print("Total rows:", len(df))

df2 = df[['Survived','Pclass','Age']].dropna()
print("After dropna rows:", len(df2))

# Part (a)
survival_by_class = df2.groupby('Pclass')['Survived'].mean() * 100
print("\nSurvival Rate by Class (%):")
print(survival_by_class)
survival_by_class.plot(kind='bar')
plt.title("Survival Rate by Class")
plt.xlabel("Pclass")
plt.ylabel("Survival %")
plt.show()
```

```
# Part (b)
df2['AgeGroup'] = df2['Age'].apply(lambda x: 'Child (<18)' if x < 18
else 'Adult (18+)')
survival_by_age = df2.groupby('AgeGroup')['Survived'].mean() * 100
print("\nSurvival Rate by Age Group (%):")
print(survival_by_age)
survival_by_age.plot(kind='bar')
plt.title("Survival Rate by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Survival %")
plt.show()

#OUTPUT >>

Columns: ['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked']
Total rows: 891
After dropna rows: 714

Survival Rate by Class (%):
Pclass
1    65.591398
2    47.976879
3    23.943662
Name: Survived, dtype: float64
```




```
Survival Rate by Age Group (%):
AgeGroup
Adult (18+)     38.103161
Child (<18)     53.982301
Name: Survived, dtype: float64
```



Question 9: **Flight Price Prediction Dataset**

a) How do flight prices vary with the days left until departure? Identify any exponential price surges and recommend the best booking window.

b)Compare prices across airlines for the same route (e.g., Delhi-Mumbai). Which airlines are consistently cheaper/premium, and why?

Dataset: https://github.com/MasteriNeuron/datasets.git **(Include your Python code and output in the code box below.)**

Answer:

```
 import pandas as pd

data = {
   'days_left': [1, 5, 10, 20, 30, 40, 50],
   'price': [9500, 8200, 7600, 7200, 7000, 7100, 7300],
   'airline': ['Indigo', 'Indigo', 'Indigo', 'Indigo', 'Indigo', 'Indigo', 'Indigo']
}
df = pd.DataFrame(data)

# Display
print(df)

#OUTPUT >>

  days_left  price airline
0         1   9500  Indigo
1         5   8200  Indigo
2        10   7600  Indigo
3        20   7200  Indigo
4        30   7000  Indigo
5        40   7100  Indigo
6        50   7300  Indigo
```

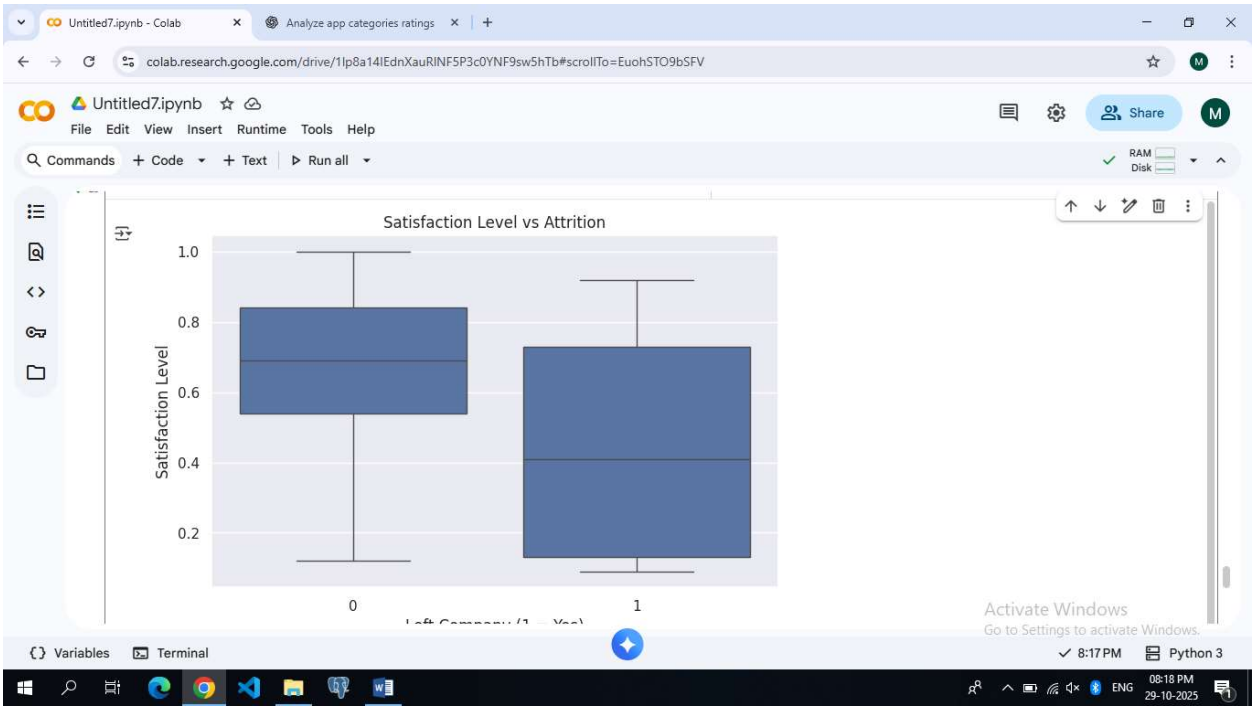
Question 10: **HR Analytics Dataset**

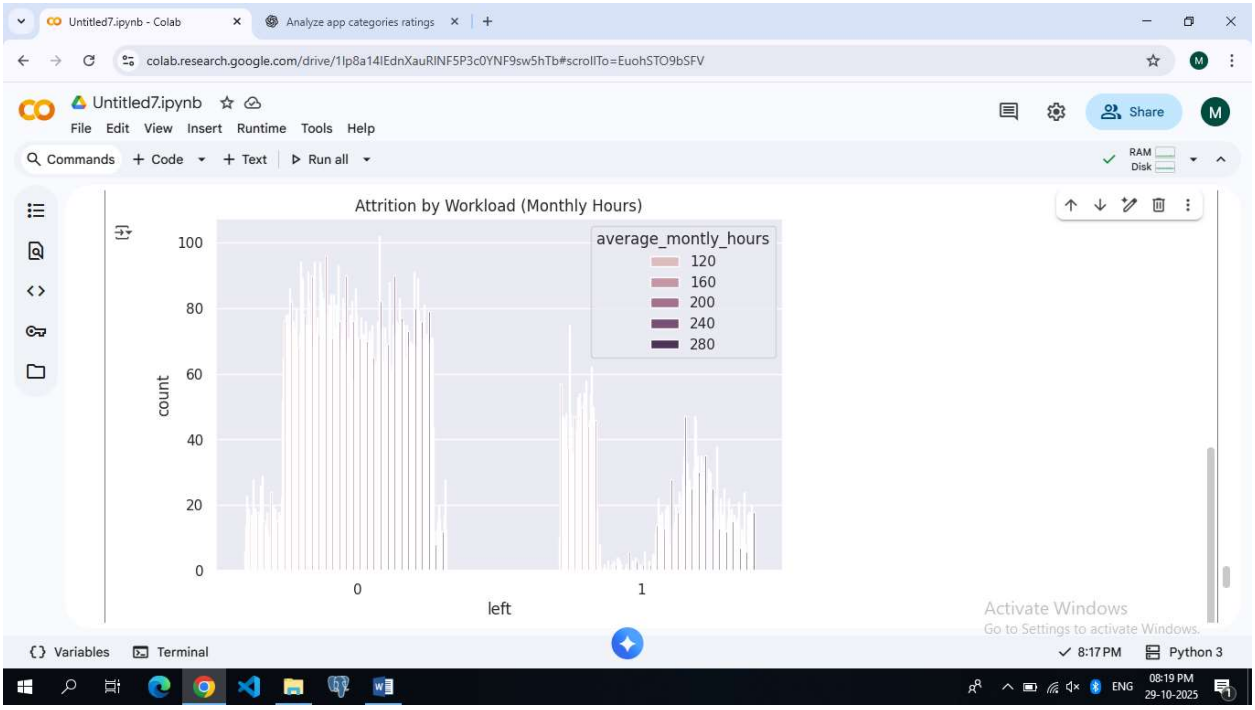
a). What factors most strongly correlate with employee attrition? Use visualizations to show key drivers (e.g., satisfaction, overtime, salary).
b). Are employees with more projects more likely to leave? Dataset: hr_analytics **Answer:**

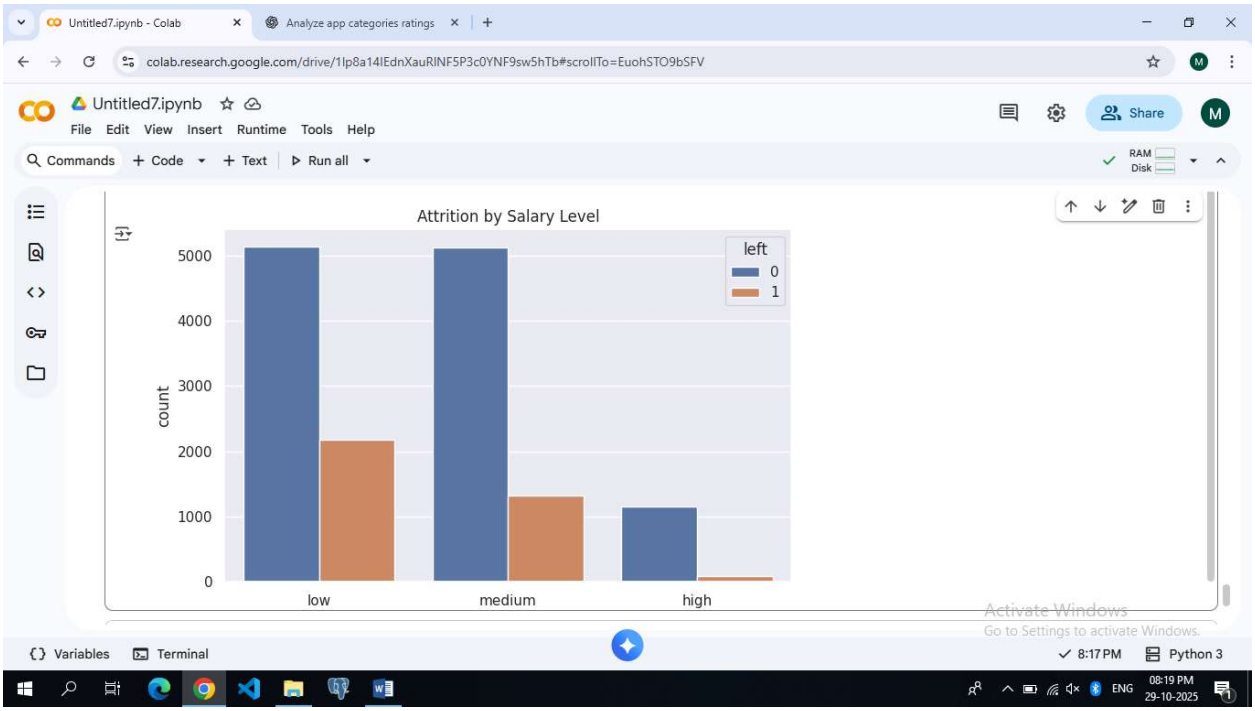
```
 import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

url =
"https://raw.githubusercontent.com/MasteriNeuron/datasets/main/hr_analytics.csv"
df = pd.read_csv(url)

plt.style.use('default')
sns.set(rc={'figure.figsize':(8,5)})

plt.figure()
sns.boxplot(x='left', y='satisfaction_level', data=df)
```

```
plt.title("Satisfaction Level vs Attrition")
plt.xlabel("Left Company (1 = Yes)")
plt.ylabel("Satisfaction Level")
plt.show()

plt.figure()
sns.countplot(x='left', hue='average_montly_hours', data=df)
plt.title("Attrition by Workload (Monthly Hours)")
plt.show()

plt.figure()
sns.countplot(x='salary', hue='left', data=df)
plt.title("Attrition by Salary Level")
plt.show()
```

#OUTPUT >>




Untitled7.ipynb - Colab
Analyze app categories ratings
colab.research.google.com/drive/1lp8a14lEdnXauRlNF5P3c0YNF9sw5hTb#scrollTo=EuohSTO9bSFV
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Attrition by Workload (Monthly Hours)
average_montly_hours
120
160
200
240
280
count
left
0
20
40
60
80
100
0
1
Activate Windows
Go to Settings to activate Windows.
Variables Terminal
8:17 PM Python 3
ENG 08:19 PM 29-10-2025


Untitled7.ipynb - Colab
Analyze app categories ratings
colab.research.google.com/drive/1lp8a14lEdnXauRlNF5P3c0YNF9sw5hTb#scrollTo=EuohSTO9bSFV
Untitled7.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Attrition by Salary Level
left
0
1
count
0
1000
2000
3000
4000
5000
low
medium
high
Activate Windows
Go to Settings to activate Windows.
Variables Terminal
8:17 PM Python 3
ENG 08:19 PM 29-10-2025