JAIPURIA INSTITUTE OF MANAGEMENT
LUCKNOW NOIDA JAIPUR INDORE

JAIPURIA INSTITUTE OF MANAGEMENT, NOIDA

PGDM / PGDM (M) / PGDM (SM)

FOURTH TRIMESTER (Batch 2023-25)

END TERM EXAMINATIONS, SEPTEMBER 2024

SET - A

Course Name	Machine Learning	Course Code	20827
Max. Time	**2 hours**	Max. Marks	**40 MM**

INSTRUCTIONS:

a. All questions are compulsory to attempt

1. Read the following case and answer the questions given at the end:

Case Study: FreshBites Gourmet Pvt. Ltd.

FreshBites Gourmet Pvt. Ltd., founded in 2020 by two food enthusiasts, Meera and Kunal, aimed to revolutionize the way people experienced healthy, gourmet food in urban areas. Based in Delhi, their company initially began as an online meal delivery service, focusing on providing nutritious, gourmet meals made from fresh, organic ingredients. Their idea was simple: offer customers a convenient, healthy, and delicious alternative to fast food.

FreshBites gained traction quickly, catering to health-conscious individuals, busy professionals, and fitness enthusiasts. They offered a subscription model for their customers, providing daily meals based on personalized dietary preferences, fitness goals, and nutritional requirements. As demand surged, FreshBites expanded their operations by opening physical outlets in major metro cities and partnering with gyms and wellness centers to offer meal services.

However, with expansion came challenges. While FreshBites grew to 75 outlets across the country and scaled up its delivery services, maintaining the same high standard of food quality, nutritional value, and timely delivery became difficult. Some customers started reporting discrepancies in portion sizes, delays in delivery, and varying taste profiles across different outlets. The core promise of personalized meals was not always being met, and customer satisfaction began to decline.

Additionally, managing inventory and sourcing high-quality organic ingredients in such large quantities posed a significant challenge. Variations in supply, fluctuations in ingredient quality, and the rise in prices of organic produce further complicated their operations. Furthermore, tracking customer preferences, feedback, and dietary data became overwhelming, as manual processes were no longer able to handle the large volume of information.

Recognizing these growing pains, Meera and Kunal turned their attention to technological solutions. They had heard about the potential of machine learning to enhance business operations and customer experience. Now, they were eager to explore how machine learning models could help FreshBites overcome its operational challenges and scale up effectively without compromising quality.

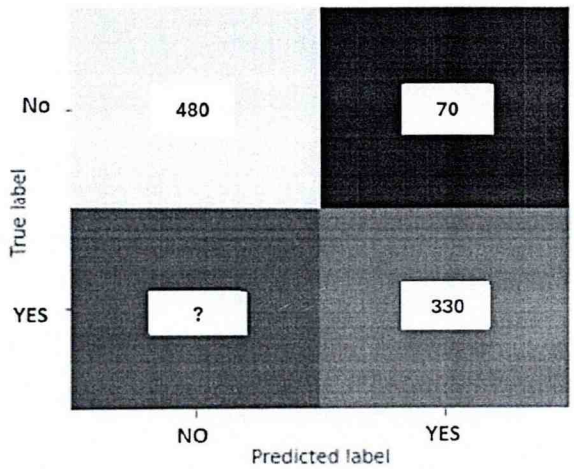
Questions

Facing the ongoing challenges with their business, Meera asked, " How can we overcome the current issues with the help of Machine Learning?" Meera and Kunal have hired you as ML analyst in their company. You are tasked with harnessing the power of data to overcome the current issues the company is facing?"

(12 Marks = 4*3)

a. Using ML technique, describe the methodology you would employ to enhance business operations and customer experience.
b. What data sources and features would you utilize, and what algorithms and models would you consider? Discuss with the help of any one challenge.
c. Following the analysis in the above questions, what data driven strategies and recommendations would you propose to Meera and Kunal?

2. Streamify, a leading subscription-based video streaming service, has noticed a significant increase in subscriber churn over the past six months, resulting in a considerable impact on its revenue and growth projections. The management team has tasked the Data Analytics Department with developing a predictive model to proactively identify subscribers at risk of churning. The objective is to better understand the key factors driving churn and implement targeted retention strategies.
After extensive data analysis and feature engineering, the analytics team built and trained a classification model using historical subscriber data. The model was subsequently validated on a test dataset. The confusion matrix for the test data is provided below. In this matrix, 'No' indicates subscribers who did not churn, and 'Yes' indicates subscribers who churned.



The total number of actual churn cases (Actual Yes) in the test dataset is known to be 450.

Questions:

A. Find the missing value in the confusion matrix. (1 Mark)

B. Based on the completed confusion matrix, determine and explain the values of the following metrics: (0.5*8 = 4 Marks)

a. True Positive (TP) - _____
b. False Positive (FP) - _____
c. False Negative (FN) - _____
d. True Negative (TN) - _____
e. Accuracy - _____
f. Recall (P) – _____
g. Precision (P) – _____
h. F1-score (P) - _____

C. How does the above metrics will help evaluate the performance of the predictive model and identify areas for improvement in accurately predicting subscriber churn? (2 Marks)

3. Jupiter Institute of Management (JIM) is one of the leading private B-school based in Maharashtra, India. The placement team of the institute believed that using machine learning models they can derive the key insights and would be able to predict the amount of salary offered. They have collected the past dataset of students of MBA and analyzed it to check their belief. The data description is given below:

Variables	Description
Roll_no	Roll number of a student
PGDM_CGPA	CGPA in PGDM out of 10
CRT_Score	Score of Student in "Comprehension refresher test" out of 100
Gender	Gender of a student
Work_Experience	Whether student has a Work experience or not
Salary_Offered	Salary Package offered to student (in thousands of Rupees)

The data has been analyzed using python and given in Appendix 1. As ML analyst you are required to develop a report to answer the following questions to better understand the students' placement. **(21 marks)**

1. Identify the number of features in the dataset. Determine the dimension of the given dataset? (1 mark)
2. Estimate the percentage of experienced students? (0.5 marks)
3. Determine the highest and lowest salary offered. Also, indicate the median salary. (1.5 marks)
4. What percentage of students were offered a salary package of more than Rs. 10,00,000? (1 mark)
5. Examine the histogram of salary offered. Comment upon its distribution. (2 marks)
6. Does their exist outliers for salary offered. Identify the outlier values (2 marks)
7. Suggest two ways for treating outlier values in the dataset. (2 marks)
8. Develop a regression equation. Estimate the predicted salary for an experienced male student who scored 7.4 CGPA and 80 marks in CRT? (2 marks)
9. In regression analysis, estimate R^2 and RMSE. Comment upon their importance. (4 marks)
10. What is the relevance of dividing the data into train and test set? (2 marks)
11. Had there been missing values in the salary offered column, how would you treat the missing values? (2 mark)
12. In your opinion, how can we improve the model fit? (1 mark)

Appendix 1

In [1]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sn
from sklearn.metrics import r2_score, mean_squared_error
```

In [2]:
```
dataset = pd.read_csv("mba_placement_ML1.csv")
dataset[0:3]
```

Out[2]:

	Roll_no	CRT_Score	PGDM_CGPA	Gender	Work_Experience	Salary_Offered
0	1	47.0	6.24	M	No	725.0
1	2	42.0	5.71	M	No	733.0
2	3	60.0	5.92	M	No	725.0

Note: Salary is in thousand of Rupees (i.e. 925 thousand = 9,25,000 per annum)

In [3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207 entries, 0 to 206
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Roll_no          207 non-null    int64
 1   CRT_Score        207 non-null    float64
 2   PGDM_CGPA        207 non-null    float64
 3   Gender           207 non-null    object
 4   Work_Experience  207 non-null    object
 5   Salary_Offered   207 non-null    float64
dtypes: float64(3), int64(1), object(2)
memory usage: 9.8+ KB
```

In [4]:
```
dataset.Gender.value_counts()
```

Out[4]:
```
Gender
M    126
F     81
Name: count, dtype: int64
```

In [5]:
```
1 dataset.Work_Experience.value_counts()
```

Out[5]:
```
Work_Experience
No     142
Yes     65
Name: count, dtype: int64
```

In [6]:
```
1 dataset[['PGDM_CGPA', 'Salary_Offered']].describe()
```

Out[6]:

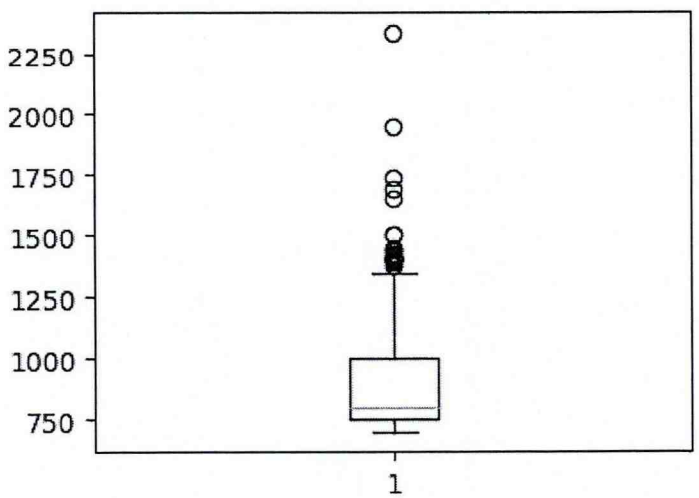
	PGDM_CGPA	Salary_Offered
count	207.000000	207.000000
mean	6.880242	930.207671
std	0.918139	270.333098
min	5.170000	700.000000
25%	6.160000	750.000000
50%	6.750000	800.000000
75%	7.500000	1000.000000
max	9.120000	2340.000000

In [7]:
```
1 dataset.groupby('Work_Experience')['Salary_Offered'].mean().reset_index()
```

Out[7]:

	Work_Experience	Salary_Offered
0	No	843.175972
1	Yes	1120.338462

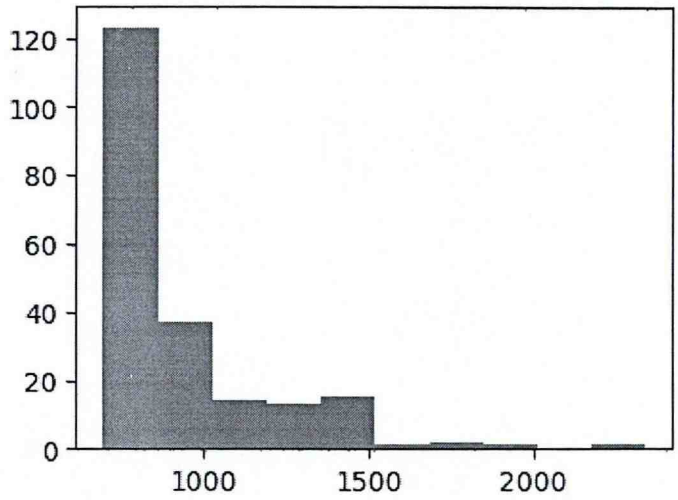
In [8]:
```
1 plt.figure(figsize=(4, 3))
2 box = plt.boxplot(dataset['Salary_Offered']);
```



In [9]:
```
[item.get_ydata()[0] for item in box['caps']]
```

Out[9]: [700.0, 1350.0]

In [10]:
```
plt.figure(figsize=(4, 3))
plt.hist( dataset['Salary_Offered'] );
```



In [11]:
```
X_features = ['CRT_Score','Gender', 'PGDM_CGPA','Work_Experience']
```

In [12]:
```
categorical_features = ['Gender', 'Work_Experience']
```

In [13]:
```
encoded_df = pd.get_dummies(dataset[X_features],
                            columns = categorical_features,
                            drop_first = True, dtype = int )
```

In [14]:
```
X = sm.add_constant(encoded_df )
Y = dataset['Salary_Offered']
```

In [15]:
```
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split( X , Y,
                                                     train_size = 0.80,
                                                     random_state = 20)
```

In [16]:
```
reg_model_1 = sm.OLS(train_y, train_X).fit()
reg_model_1.summary2()
```

Out[16]:

Model:	OLS	Adj. R-squared:	0.598
Dependent Variable:	Salary_Offered	AIC:	2168.3361
Date:	2024-09-20 13:47	BIC:	2183.8658
No. Observations:	165	Log-Likelihood:	-1079.2
Df Model:	4	F-statistic:	62.06
Df Residuals:	160	Prob (F-statistic):	1.42e-31
R-squared:	0.608	Scale:	28961.

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	-416.1740	107.8971	-3.8571	0.0002	-629.2602	-203.0878
CRT_Score	4.7577	1.2606	3.7742	0.0002	2.2682	7.2473
PGDM_CGPA	135.4027	16.9385	7.9938	0.0000	101.9508	168.8545
Gender_M	49.1515	27.1542	1.8101	0.0722	-4.4754	102.7784
Work_Experience_Yes	179.6892	30.2669	5.9368	0.0000	119.9151	239.4634

Omnibus:	43.392	Durbin-Watson:	2.134
Prob(Omnibus):	0.000	Jarque-Bera (JB):	123.570
Skew:	1.040	Prob(JB):	0.000
Kurtosis:	6.694	Condition No.:	565

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [17]:
```
encoded_df = encoded_df.drop('Gender_M', axis=1)
```

In [18]:
```
X = sm.add_constant(encoded_df)
Y = dataset['Salary_Offered']
```

In [19]:
```
from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split( X ,Y,
                                                     train_size = 0.80,
                                                     random_state = 20 )
```

In [20]:
```
reg_model_2 = sm.OLS(train_y, train_X).fit()
reg_model_2.summary2()
```

Out[20]:

Model:	OLS	Adj. R-squared:	0.593
Dependent Variable:	Salary_Offered	AIC:	2169.6808
Date:	2024-09-20 13:47	BIC:	2182.1046
No. Observations:	165	Log-Likelihood:	-1080.8
Df Model:	3	F-statistic:	80.51
Df Residuals:	161	Prob (F-statistic):	7.26e-32
R-squared:	0.600	Scale:	29371.

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	-396.4377	108.1010	-3.6673	0.0003	-609.9164	-182.9590
CRT_Score	4.7994	1.2693	3.7812	0.0002	2.2928	7.3060
PGDM_CGPA	136.2565	17.0512	7.9910	0.0000	102.5836	169.9293
Work_Experience_Yes	182.8881	30.4281	6.0105	0.0000	122.7984	242.9778

Omnibus:	46.500	Durbin-Watson:	2.115
Prob(Omnibus):	0.000	Jarque-Bera (JB):	128.843
Skew:	1.132	Prob(JB):	0.000
Kurtosis:	6.690	Condition No.:	562

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [21]:
```
pred_y = reg_model_2.predict( test_X )
pred_y = np.round(pred_y, 2)
```

In [22]:
```
pred_df = pd.DataFrame({'CRT_Score': test_X['CRT_Score'],
                        'PGDM_CGPA': test_X['PGDM_CGPA'],
                        'Work_Experience_Yes': test_X['Work_Experience_Yes
                        'Actual Y':  test_y,
                        'predicted Y': pred_y,
                        })
pred_df[0:3]
```

Out[22]:

	CRT_Score	PGDM_CGPA	Work_Experience_Yes	Actual Y	predicted Y
133	61.0	6.33	0	750.0	758.83
5	49.0	5.87	0	740.0	638.56
13	43.0	7.17	0	760.0	786.90

In [23]:

```
r2 = np.abs(r2_score(test_y, pred_y))
rmse = np.sqrt(mean_squared_error(test_y, pred_y))
print(np.round(r2,2))
print(np.round(rmse, 2))
```

```
0.41
212.68
```