IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING
(An Autonomous Institute affiliated to JNTUH, Hyderabad)
Dundigal, Hyderabad - 500 043

LABORATORY WORK BOOK

ame of the Student ..... HIMAKAR CHAPPIDI .....

lass ..... CSE-B ..... Semester ..... VI .....

Roll Number									
2	1	9	5	1	A	0	5	6	5

ourse Code : ACIC08 ..... Course Name : DMKD Laboratory

ame of the Course Faculty ..... Dr. D DURGA BHAVANI ..... Faculty ID : IARE 10921

xercise Number : ..... Week Number : ..... 07 ..... Date : 0

S. No.	Exercise Number	EXERCISE NAME	MARKS AWARDED					
			Aim/ Preparation	Algorithm / Procedure Performance in the Lab	Source Code Calculations and Graphs	Program Execution Results and Error Analysis	Viva - Voce	Total
			4	4	4	4	4	20
1	7.1	Explore data & visualize each attribute	4	4	4	4	4	20
2								
3	7.2	Predict the test set results and find the accuracy of the model						
4								
5	7.3	visualize the confusion matrix						
6	7.4	compute Precision recall, F-measure and support						
7								
8								
9								
10								
11								
12								

HC
Signature of the Student

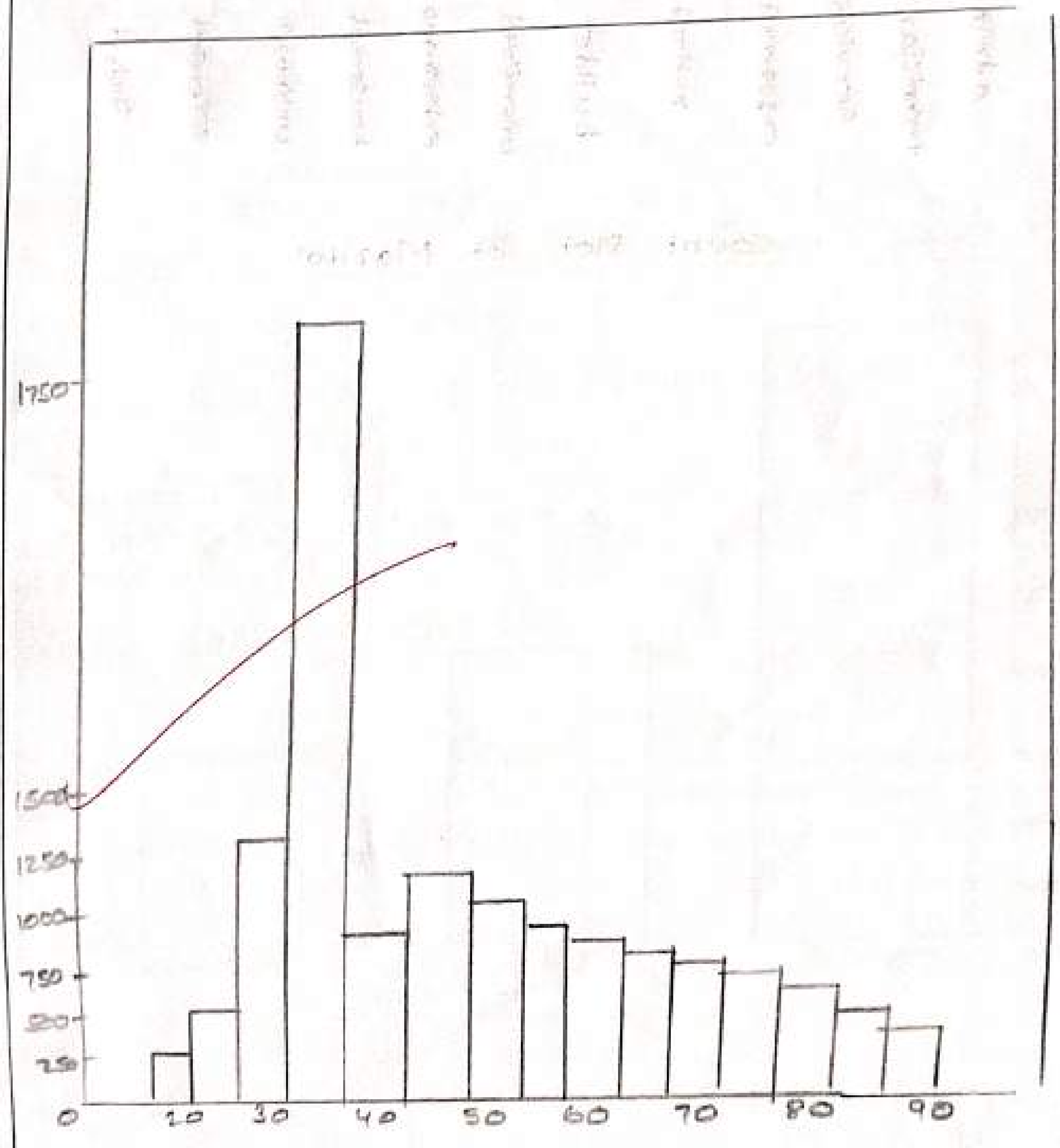
Signature of the Faculty

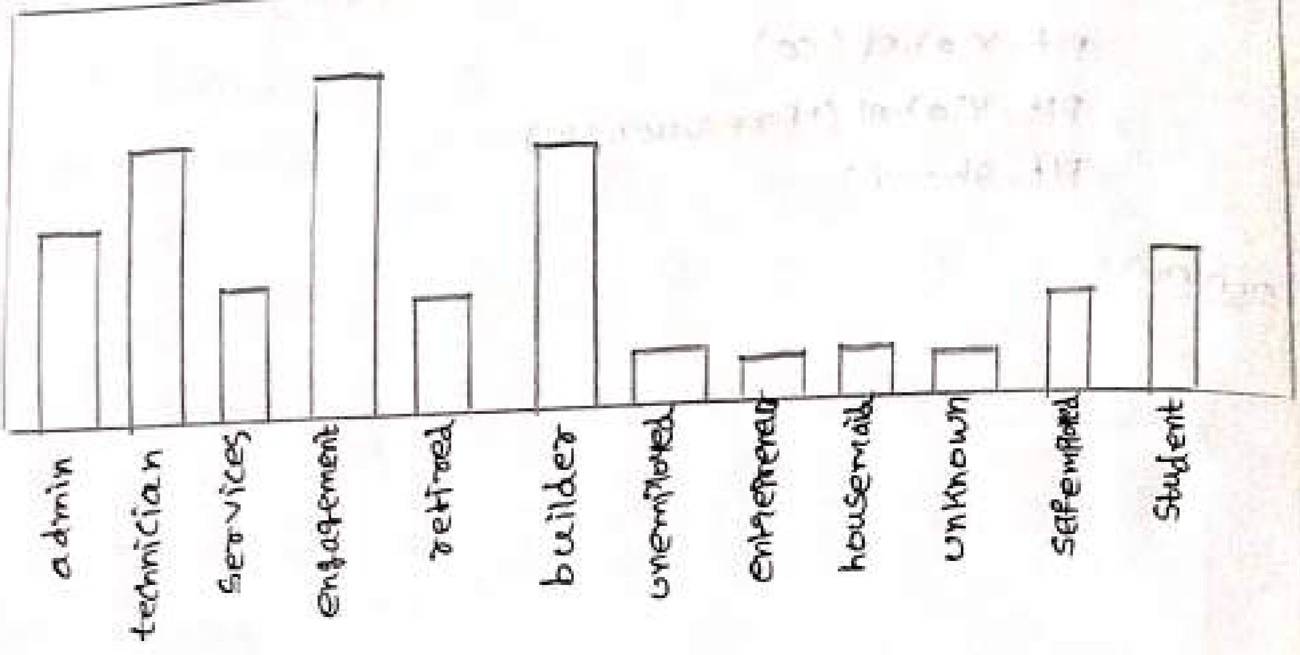
7.1 Explore data and visualize each attribute

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
bd = pd.read-csv('bank.csv")
print('First few rows of dataset:')
print (bd.head())
print('Dataset Info: ')
print (bd.info())
print('summary statistics: ')
print(bd.describe())
print('Missing values. ')
print(bd. isnull().sum())
for co in bd.columns:
  if bd[co].dtype == 'object':
     plt.figure (figsize = (8,6))
     sns.countplot (x = column, data = bd, Palette='set2')
     plt.title(f'Countplot of {co}')
     plt.xlabel (co)
     plt.ylabel ('count')
     plt.xticks(rotation = 45)
     plt.show()
  else:
     plt.figure(figsize = (8,6))
     sns.histplot (bd[co], kde = True, color = 'skyblue',
                                         bins = 20)
```

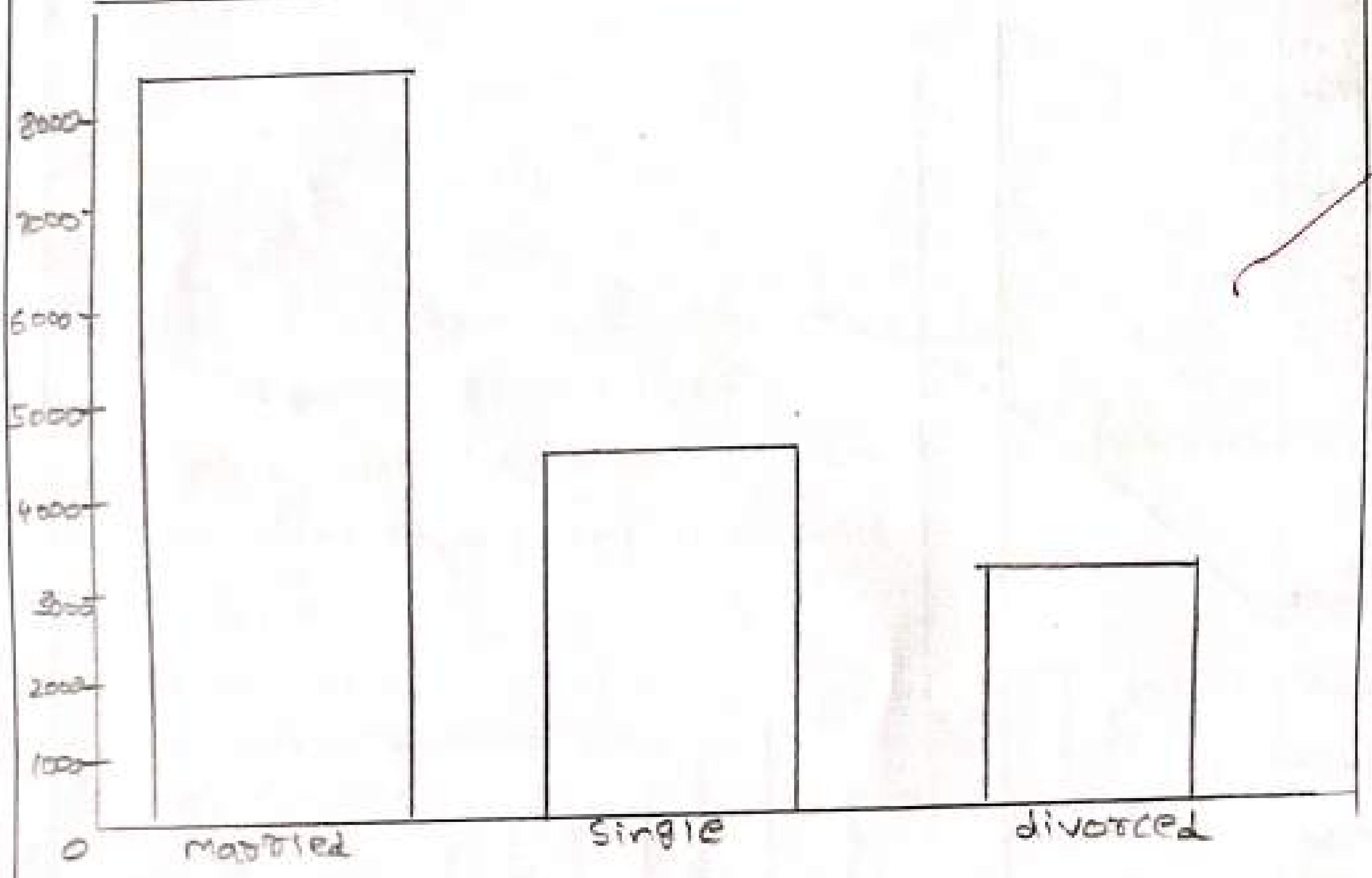
```
plt.title (f'Histogram of {co}')
plt.xlabel(co)
plt.ylabel('Frequency')
plt.show()
```

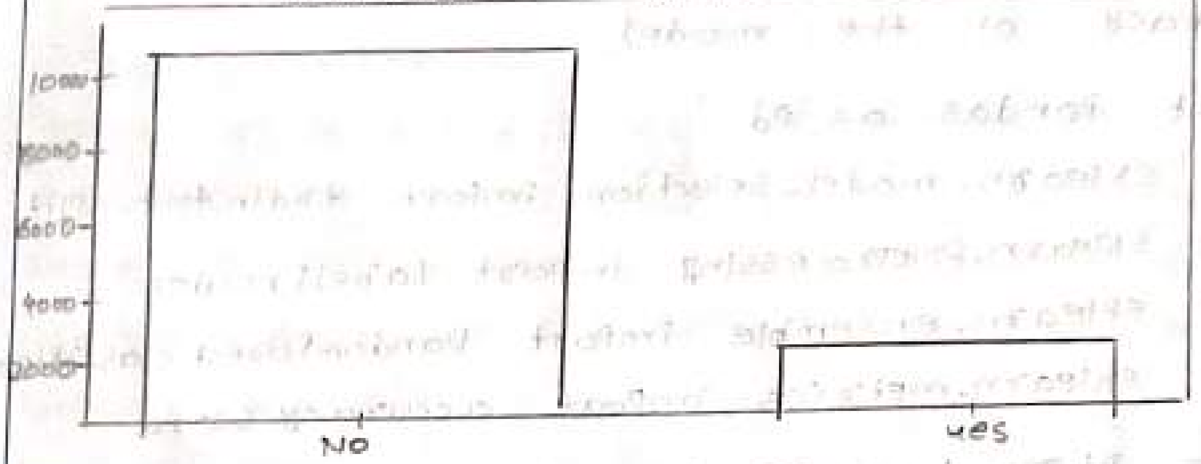
OUTPUT:



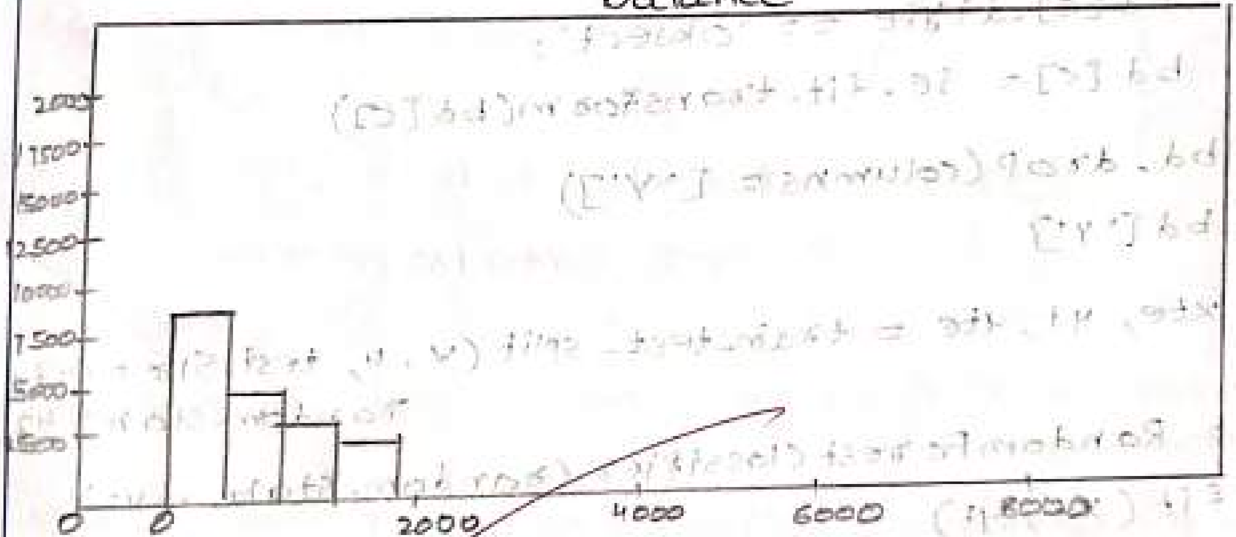


Count Plot of Marital

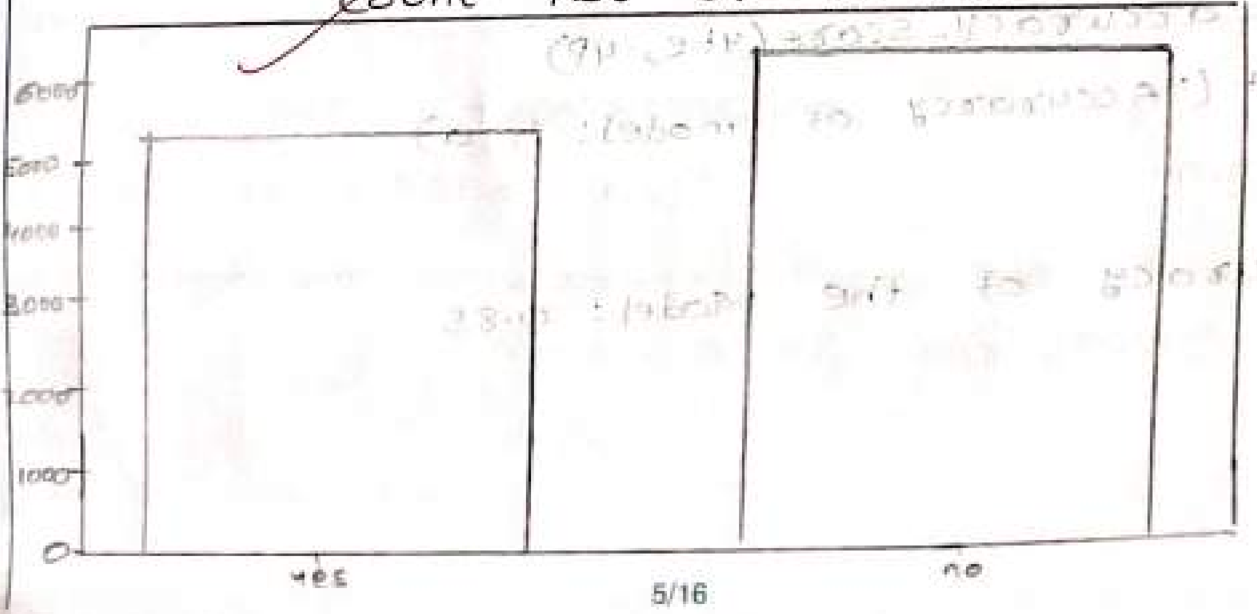




Count Plot of default



Histogram of balance



Count plot of housing

7.2 Predict the test set results and find the accuracy of the model

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForest classifier
from sklearn.metrics import accuracy_score

bd = pd.read_csv('bank.csv')
le = LabelEncoder()
for c in bd.columns:
   if bd[c].dtype == 'object':
      bd[c] = le.fit_transform(bd[c])
X = bd.drop(columns = ['Y'])
y = bd['Y']
xt, xte, yt, yte = train_test_split(x, y, test_size=0.2,
                                      random_state=42)
rfc = RandomForestClassifier(random_state =42)
rfc.fit(xt, yt)
yp = rfc.predict(xte)
a = accuracy_score(yte, yp)
print('Accurarcy of model: ', a)
```

OUTPUT:

```
Accuracy of the Model: 0.85
```

7.3 Visualize the Confusion matrix

```
import pandas as pd
from sklearn.model-selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
bd = pd.read_csv('bank.csv')
le = LabelEncoder()
for c in bd.columns:
    if bd[c].dtype == 'object':
        bd[c] = le.fit_transform(bd[c])
x = bd.drop(columns=['y'])
y = bd['y']
xt, xte, yt, yte = train_test_split(x, y, test_size=0.2
                                    random_state=42)
rfc = RandomForestClassifier(random_state=42)
rfc.fit(xt, yt)
yp = rfc.predict(xte)
cm = confusion_matrix(yte, yp)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot = True, fmt = 'd',
            cmap = 'Blues', cbar = False)
```

```
Plt.title('confusion Matrix')
Plt.xlabel('Predicted label')
Plt.ylabel('True Label')
Plt.show()
```

<u>OUTPUT:</u>

		Predicted	
		Positive	Negative
Actual	Positive	20	2
	Negative	8	1

7.4 Compute precision, recall, F-measure + support

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
bd = pd.read_csv('bank.csv')
le = LabelEncoder()
for c in bd.columns:
    if bd[c].dtype == 'object':
        bd[c] = le.fit_transform(bd[c])
x = bd.drop(columns=['y'])
y = bd['y']
xt, xte, yt, yte = train_test_split(x, y, test_size=0
                                    random_state=42)
rfc = RandomForestClassifier(random_state=42)
rfc.fit(xt, yt)
yp = rfc.predict(xte)
r = classification_report(yte, yp)
print('Classification Report:')
print(r)
```