AI Lab Week 6

Session: 2022 – 2026

Submitted by:

Muhammad Saad Akmal 2022-CS-148

Submitted To:

Sir Nauman Shafi

Department of Computer Science

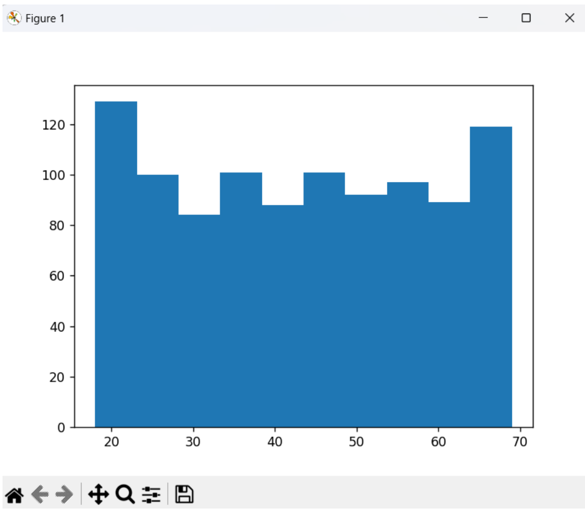
University of Engineering and Technology

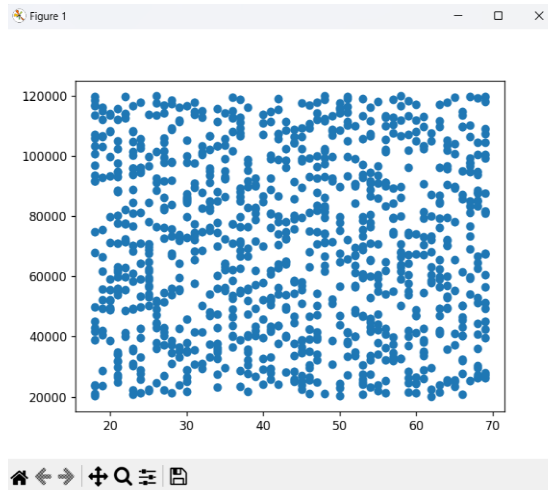
Lahore Pakistan

Case-Study 1:

```python
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from sklearn.model_selection import train_test_split


#1.2
np.random.seed(0)

rows = 1000

customer_id = [i for i in range(rows)]
gender_arr = ['Male' , 'Female']

age = np.random.randint(18,70,rows)
annual_income = np.random.randint(20000,120000,rows)
gender = np.random.choice(gender_arr , rows)

isPurchased = np.random.randint(0,2,rows)

customer_data_gen = pd.DataFrame({
    'customer_id' : customer_id,
    'age' : age,
    'gender' : gender,
    'annual_income' : annual_income,
    'purchased' : isPurchased
})

customer_data_gen.to_excel('customer_data.xlsx' , index=False)


customer_data = pd.read_excel('customer_data.xlsx')

#1.3
first_ten = customer_data.head(10)
print(f'First ten rows\n {first_ten}')
print(f'Null values\n {customer_data['annual_income'].isnull()}')

#1.4
customer_data['annual_income'].fillna(customer_data['annual_income'].median())
print(f'Filling null values\n {customer_data['annual_income']}')


#1.5
customer_data['gender_numerical'] = customer_data['gender'].map({'Male':0 , 'Female':1})
print(f'Converted data \n {customer_data}')

#1.6
customer_data['min_max_age'] = (customer_data['age'] - customer_data['age'].min()) / (customer_data['age'].max() - customer_data['age'].min())
customer_data['min_max_annual_income'] = (customer_data['annual_income'] - customer_data['annual_income'].min()) / (customer_data['annual_income'].max() - customer_data['annual_income'].min())

print(f'Normalized Ages :\n {customer_data['min_max_age']}')
print(f'Normalized Annual Income :\n {customer_data['min_max_annual_income']}')

#1.7
plt.hist(customer_data['age'])
plt.show()


plt.scatter(customer_data['age'] , customer_data['annual_income'])
plt.show()



#1.8
coorelation = customer_data[['age' , 'annual_income' , 'purchased']].corr()

print(f'Coorelation {coorelation}')


#1.9
customer_data['income_per_age'] = customer_data['annual_income'] / customer_data['age']

print(f'Income per age\n {customer_data['income_per_age']}')



#1.10
customer_data.drop('customer_id' , axis=1 , inplace=True)

print(f'Data after droped columns\n {customer_data}')


X_train, X_test = train_test_split(
    customer_data , random_state=104,test_size=0.20, shuffle=True)


print('X_train : ')
print(X_train.head())

print('')
print('X_test : ')
print(X_test.head())
```

Output:





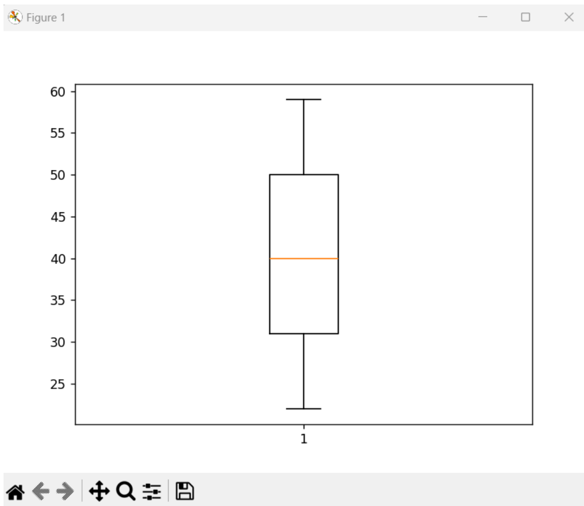

```
First ten rows
   customer_id  age  gender  annual_income  purchased
0            0   62  Female          73478          0
1            1   65    Male          65560          0
2            2   18  Female          91592          0
3            3   21  Female          88246          1
4            4   21  Female         105022          1
5            5   57  Female          34591          0
6            6   27  Female          43059          0
7            7   37  Female          62756          0
8            8   39  Female          92554          0
9            9   68  Female         104554          1
Null values
 0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: annual_income, Length: 1000, dtype: bool
Filling null values
 0       73478
1       65560
2       91592
3       88246
4      105022
        ...
995     45898
996     75483
997     96511
998    108009
999     36243
Name: annual_income, Length: 1000, dtype: int64
Converted data
     customer_id  age  gender  annual_income  purchased  gender_numerical
0              0   62  Female          73478          0                 1
1              1   65    Male          65560          0                 0
2              2   18  Female          91592          0                 1
3              3   21  Female          88246          1                 1
4              4   21  Female         105022          1                 1
..           ...  ...     ...            ...        ...               ...
995          995   54  Female          45898          0                 1
996          996   19    Male          75483          1                 0
997          997   47  Female          96511          0                 1
998          998   23    Male         108009          0                 0
999          999   34    Male          36243          1                 0

[1000 rows x 6 columns]
Normalized Ages :
 0      0.862745
1      0.921569
2      0.000000
3      0.058824
4      0.058824
         ...
995    0.705882
996    0.019608
997    0.568627
998    0.098039
999    0.313725
Name: min_max_age, Length: 1000, dtype: float64
Normalized Annual Income :
 0      0.534368
1      0.455054
```

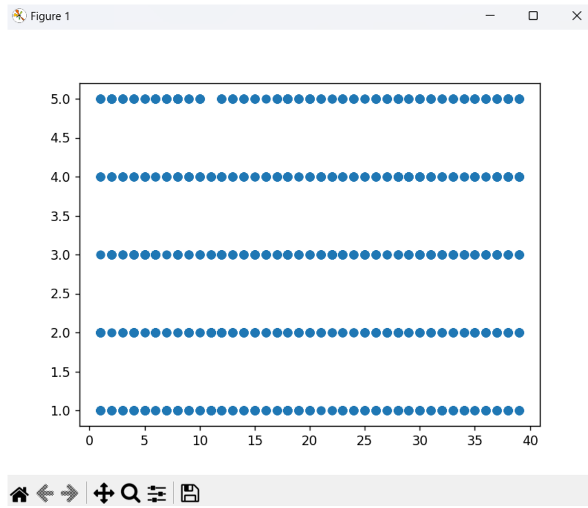
```
Normalized Annual Income :
 0      0.534368
1      0.455054
2      0.715813
3      0.682296
4      0.850339
         ...
995    0.258104
996    0.554451
997    0.765085
998    0.880259
999    0.161391
Name: min_max_annual_income, Length: 1000, dtype: float64
Coorelation                     age  annual_income  purchased
age            1.000000      -0.028314  -0.010179
annual_income -0.028314       1.000000  -0.043151
purchased     -0.010179      -0.043151   1.000000
Income per age
 0      1185.129032
1      1008.615385
2      5088.444444
3      4202.190476
4      5001.047619
          ...
995     849.962963
996    3972.789474
997    2053.425532
998    4696.043478
999    1065.970588
Name: income_per_age, Length: 1000, dtype: float64
Data after droped columns
     age  gender  annual_income  purchased  gender_numerical  min_max_age  min_max_annual_income  income_per_age
0     62  Female          73478          0                 1     0.862745               0.534368     1185.129032
1     65    Male          65560          0                 0     0.921569               0.455054     1008.615385
2     18  Female          91592          0                 1     0.000000               0.715813     5088.444444
3     21  Female          88246          1                 1     0.058824               0.682296     4202.190476
4     21  Female         105022          1                 1     0.058824               0.850339     5001.047619
..   ...     ...            ...        ...               ...          ...                    ...             ...
995   54  Female          45898          0                 1     0.705882               0.258104      849.962963
996   19    Male          75483          1                 0     0.019608               0.554451     3972.789474
997   47  Female          96511          0                 1     0.568627               0.765085     2053.425532
998   23    Male         108009          0                 0     0.098039               0.880259     4696.043478
999   34    Male          36243          1                 0     0.313725               0.161391     1065.970588

[1000 rows x 8 columns]
X_train :
     age  gender  annual_income  purchased  gender_numerical  min_max_age  min_max_annual_income  income_per_age
109   49    Male          38670          1                 0     0.607843               0.185702      789.183673
770   55  Female          23869          1                 1     0.725490               0.037443      433.981818
797   27  Female          22170          1                 1     0.176471               0.020424      821.111111
554   37    Male          68579          0                 0     0.372549               0.485295     1853.486486
951   27    Male          40428          0                 0     0.176471               0.203312     1497.333333

X_test :
     age  gender  annual_income  purchased  gender_numerical  min_max_age  min_max_annual_income  income_per_age
808   58  Female          32683          1                 1     0.784314               0.125731      563.500000
83    30    Male          84433          1                 0     0.235294               0.644102     2814.433333
643   47  Female          92536          1                 1     0.568627               0.725268     1968.851064
895   62    Male         104913          0                 0     0.862745               0.849247     1692.145161
536   38    Male          69281          1                 0     0.392157               0.492327     1823.184211
```

Case-Study 2:

```python
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from sklearn.model_selection import train_test_split



#2.2
np.random.seed(0)

rows = 1000

employee_id = [i for i in range(rows)]
gender_arr = ['Male' , 'Female']

age = np.random.randint(22,60,rows)
years_of_experience = np.random.randint(1,40,rows)

gender = np.random.choice(gender_arr , rows)
performance_rating = np.random.randint(1,6,rows)


employee_data_gen = pd.DataFrame({
    'employee_id' : employee_id,
    'age' : age,
    'gender' : gender,
    'experience' : years_of_experience,
    'performance_rating' : performance_rating
})

employee_data_gen.to_excel('employee_data.xlsx' , index=False)

employee_data = pd.read_excel('employee_data.xlsx')


#2.3
first_fifteen = employee_data.head(15)
print(f'First 15 rows \n {first_fifteen}')
print(f'Null values\n {employee_data.isnull()}')


#2.4
employee_data['experience'].fillna(employee_data['experience'].mean())
print(f'Filling null values {employee_data['experience']}')


#2.5
employee_data['numerical_gender'] = employee_data['gender'].map({'Male':0 , 'Female':1})
print(f'Numerical column of gender\n {employee_data['numerical_gender']}')


#2.6
outliers = np.where((employee_data['experience'] > 40))
print(f'Outliers\n {outliers}')


#2.7
employee_data['age_normalized'] = (employee_data['age'] - employee_data['age'].mean()) / employee_data['age'].std()
employee_data['experience_normalized'] = (employee_data['experience'] - employee_data['experience'].mean()) / employee_data['experience'].std()
print(f'Normalized Data\n {employee_data}')


#2.8
plt.boxplot(employee_data['age'])
plt.show()


plt.scatter(employee_data['experience'] , employee_data['performance_rating'])
plt.show()


#2.9
coorelation = employee_data[['age' , 'performance_rating' , 'experience']].corr()
print(f'Coorelation {coorelation}')


#2.10
employee_data['experience_per_age'] = employee_data['experience'] / employee_data['age']
print(f'Experience per age\n {employee_data['experience_per_age']}')


#2.11
employee_data.drop('employee_id' , axis=1 , inplace=True)

print(f'Data after droped columns\n {employee_data}')

X_train, X_test = train_test_split(
  employee_data , random_state=104,test_size=0.20, shuffle=True)

print('X_train : ')
print(X_train.head())

print('')
print('X_test : ')
print(X_test.head())
```

Output:






```
First 15 rows
    employee_id  age  gender  experience  performance_rating
0             0   22    Male           8                   5
1             1   25  Female          17                   4
2             2   25    Male          35                   3
3             3   31    Male           5                   1
4             4   41  Female          27                   2
5             5   43    Male          37                   4
6             6   58  Female           7                   1
7             7   45    Male          33                   4
8             8   28  Female          18                   4
9             9   46  Female          38                   5
10           10   46  Female          28                   4
11           11   34    Male          15                   2
12           12   23  Female          23                   3
13           13   45  Female          32                   3
14           14   46  Female           9                   4
Null values
     employee_id    age  gender  experience  performance_rating
0          False  False   False       False               False
1          False  False   False       False               False
2          False  False   False       False               False
3          False  False   False       False               False
4          False  False   False       False               False
..           ...    ...     ...         ...                 ...
995        False  False   False       False               False
996        False  False   False       False               False
997        False  False   False       False               False
998        False  False   False       False               False
999        False  False   False       False               False

[1000 rows x 5 columns]
Filling null values 0        8
1      17
2      35
3       5
4      27
       ..
995    13
996    18
997    32
998    38
999     1
Name: experience, Length: 1000, dtype: int64
Numerical column of gender
 0      0
1      1
2      0
3      0
4      1
      ..
995    0
996    0
997    1
998    0
999    1
Name: numerical_gender, Length: 1000, dtype: int64
Outliers
 (array([], dtype=int64),)
```

```
Normalized Data
     employee_id  age  gender  experience  performance_rating  numerical_gender  age_normalized  experience_normalized
0              0   22    Male           8                   5                 0       -1.656915              -1.089687
1              1   25  Female          17                   4                 1       -1.385453              -0.290275
2              2   25    Male          35                   3                 0       -1.385453               1.308548
3              3   31    Male           5                   1                 0       -0.842528              -1.356157
4              4   41  Female          27                   2                 1        0.062346               0.597960
..           ...  ...     ...         ...                 ...               ...             ...                    ...
995          995   39    Male          13                   3                 0       -0.118629              -0.645569
996          996   25    Male          18                   1                 0       -1.385453              -0.201452
997          997   31  Female          32                   4                 1       -0.842528               1.042077
998          998   51    Male          38                   5                 0        0.967220               1.575018
999          999   25  Female           1                   2                 1       -1.385453              -1.711451

[1000 rows x 8 columns]
Coorelation                         age  performance_rating  experience
age                 1.000000          -0.003395    0.016177
performance_rating -0.003395           1.000000    0.021820
experience          0.016177           0.021820    1.000000
Experience per age
 0      0.363636
1      0.680000
2      1.400000
3      0.161290
4      0.658537
         ...
995    0.333333
996    0.720000
997    1.032258
998    0.745098
999    0.040000
Name: experience_per_age, Length: 1000, dtype: float64
Data after droped columns
     age  gender  experience  performance_rating  numerical_gender  age_normalized  experience_normalized  experience_per_age
0     22    Male           8                   5                 0       -1.656915              -1.089687            0.363636
1     25  Female          17                   4                 1       -1.385453              -0.290275            0.680000
2     25    Male          35                   3                 0       -1.385453               1.308548            1.400000
3     31    Male           5                   1                 0       -0.842528              -1.356157            0.161290
4     41  Female          27                   2                 1        0.062346               0.597960            0.658537
..   ...     ...         ...                 ...               ...             ...                    ...                 ...
995   39    Male          13                   3                 0       -0.118629              -0.645569            0.333333
996   25    Male          18                   1                 0       -1.385453              -0.201452            0.720000
997   31  Female          32                   4                 1       -0.842528               1.042077            1.032258
998   51    Male          38                   5                 0        0.967220               1.575018            0.745098
999   25  Female           1                   2                 1       -1.385453              -1.711451            0.040000

[1000 rows x 8 columns]
X_train :
     age  gender  experience  performance_rating  numerical_gender  age_normalized  experience_normalized  experience_per_age
109   32    Male          18                   5                 0       -0.752041              -0.201452            0.562500
770   59    Male          30                   2                 0        1.691119               0.864430            0.508475
797   46  Female          38                   3                 1        0.514783               1.575018            0.826087
554   51  Female          33                   1                 1        0.967220               1.130901            0.647059
951   40  Female          26                   1                 1       -0.028142               0.509136            0.650000

X_test :
     age  gender  experience  performance_rating  numerical_gender  age_normalized  experience_normalized  experience_per_age
808   33  Female          11                   1                 1       -0.661553              -0.823216            0.333333
83    52    Male          22                   4                 0        1.057707               0.153842            0.423077
643   27  Female          38                   2                 1       -1.204478               1.575018            1.407407
895   56  Female           5                   5                 1        1.419657              -1.356157            0.089286
536   40    Male          25                   5                 0       -0.028142               0.420313            0.625000
```