EXPERIMENT - 3

AIM: Perform Data Modeling.

TO - DO:

- Partition the data set, for example 75% of the records are included in the training data set and 25% are included in the test data set.
- Use a bar graph and other relevant graphs to confirm your proportions.
- Identify the total number of records in the training data set.
- Validate partition by performing a two-sample Z-test.

ABOUT DATASET:

Link of our dataset: https://www.kaggle.com/datasets/shivan118/churn-modeling-dataset

This data set contains details of a bank's customers and the target variable is a binary variable reflecting the fact whether the customer left the bank (closed his account) or he continues to be a customer.

Columns: [Row Number, Customer Id, Surname,CreditScore,Geaography, gender,Age,Tenure,Balance,NumOfProducts, HasCrCard, IsActiveMember,EstimatedSalary,exited]

THEORY:

Data Partitioning:

Data partitioning in data mining is the division of the whole data available into two or three non-overlapping sets: the training set , the validation set , and the test set . If the data set is very large, often only a portion of it is selected for the partitions. Partitioning is normally used when the model for the data at hand is being chosen from a broad set of models. The basic idea of data partitioning is to keep a subset of available data out of analysis, and to use it later for verification of the model. Data partitioning is normally used in supervised learning techniques in data mining where a predictive model is chosen from a set of models, using their performance on the training set as the validation of choice. Some examples of such techniques are classification trees , regression trees , neural networks , and nonlinear variants of the discriminant analysis .

For example, a researcher developed a method for prediction of time series of stock prices data. The parameters of the model have been fitted to the available data, and the model demonstrates high prediction accuracy on these data. But this does not necessarily mean that the model will predict new data that well -- the model has been especially tuned to the characteristics (including random chance aspects) of the data used to fit it. Data partitioning is used to avoid such overly optimistic estimates of the model precision.

Hypothesis testing:

A **hypothesis** is often described as an “educated guess” about a specific parameter or population. Once it is defined, one can collect data to determine whether it provides enough evidence that the hypothesis is true.
Hypothesis testing is an act in statistics whereby an analyst tests an assumption regarding a population parameter. The methodology employed by the analyst depends on the nature of the data used and the reason for the analysis. Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data. Such data may come from a larger population, or from a data-generating process. The word "population" will be used for both of these cases in the following descriptions.

- Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data.
- The test provides evidence concerning the plausibility of the hypothesis, given the data.
- Statistical analysts test a hypothesis by measuring and examining a random sample of the population being analyzed.

SCREENSHOTS:

- Partition the data set, for example 75% of the records are included in the training data set and 25% are included in the test data set.

```
#loading the dataset
import pandas as pd
data = pd.read_csv("Churn_Modelling.csv")
data
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

```
#Encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
data['Age'] = le.fit_transform(data['Age'])
data
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	24	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	23	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	24	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	21	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	25	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	21	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	17	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	18	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	24	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	10	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

```
[ ] #splitting the data
    from sklearn.model_selection import train_test_split
    X= data.drop(columns=['Age'])
    Y = data['Age']
    x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size=0.25)
```

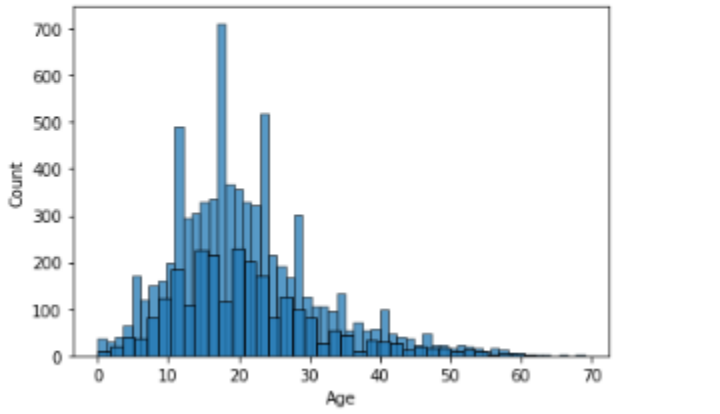
```
[ ] #the train and test data has been splitted into 75% and 25% respectively
    print(x_train.shape)
    print(x_test.shape)
```

```
(7500, 13)
(2500, 13)
```

- Use a bar graph and other relevant graphs to confirm your proportions.

```
[7] import seaborn as sns
    sns.histplot(y_train)
    sns.histplot(y_test)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9003a4ce50>
```



- Identify the total number of records in the training data set.

```
[ ] print(x_train.shape)
    print(y_train.shape)
```

```
(7500, 13)
(7500,)
```

```
[ ] x_train.value_counts()
```

```
RowNumber  CustomerId  Surname          CreditScore  Geography  Gender  Tenure  Balance    NumOfProducts  HasCrCard  IsActiveMember  EstimatedSalary  Exited
1          15634602    Hargrave         619          France     Female  2       0.00       1              1          1               101348.88        1         1
6638       15668767    Kenenna          850          France     Male    3       0.00       2              1          0               195033.07        0         1
6650       15635277    Coates           605          Spain      Male    7       142643.54  1              1          0               189310.27        0         1
6649       15731751    Osinachi         437          France     Female  1       120923.52  1              0          1               78854.57         0         1
6648       15691627    Tai              713          France     Female  8       0.00       1              1          1               16403.41         0         1
                                                                                                                                                      ..
3259       15577514    Mai              698          Germany    Female  7       121263.62  1              1          1               13387.88         0         1
3258       15709183    Davidson         707          France     Female  3       102346.86  1              1          1               114672.64        0         1
3257       15573926    Lung             735          Spain      Male    7       86131.71   2              0          0               93478.96         0         1
3256       15671387    Fetherstonhaugh  507          France     Female  4       89349.47   2              0          0               180626.68        0         1
10000      15628319    Walker           792          France     Female  4       130142.79  1              1          0               38190.78         0         1
Length: 7500, dtype: int64
```

```
[ ] y_train.value_counts()
```

```
19    359
20    356
17    350
18    349
16    348
     ...
69      1
66      1
68      1
65      1
64      1
Name: Age, Length: 70, dtype: int64
```

```
[ ] x_train.count()
```

```
RowNumber          7500
CustomerId         7500
Surname            7500
CreditScore        7500
Geography          7500
Gender             7500
Tenure             7500
Balance            7500
NumOfProducts      7500
HasCrCard          7500
IsActiveMember     7500
EstimatedSalary    7500
Exited             7500
dtype: int64
```

```
[ ] y_train.count()
```

```
7500
```

- Validate partition by performing a two-sample Z-test.

```
[ ]  #Null hypothesis = There is no significant difference between test and train datset
     #Alternative hypothesis = There is significant difference between the two datasets
```

```
[ ]  from statsmodels.stats.weightstats import ztest as ztest
     ztest(y_test,y_train,value=0)
```

```
     (-0.9044024505967269, 0.36578203921585084)
```

```
[ ]  def results(p):
       if (p['p_value']<0.05):p['hypothesis_accepted'] = 'alternative'
       if (p['p_value']>=0.05):p['hypothesis_accepted'] = 'null'

       df  = pd.DataFrame(p,index=[''])
       cols = ['value_1','value_2','score','p_value','hypothesis_accepted']
       return df[cols]
```

```
[ ]  p = { } #dictionary to store results
     p['value_1'],p['value_2'] = y_train.mean(),y_test.mean()
     p['score'],p['p_value'] = ztest(y_train,y_test,alternative='two-sided')
     results(p)
```

	value_1	value_2	score	p_value	hypothesis_accepted
	20.975333	20.7564	0.904402	0.365782	null

Since the Hypothesis accepted is null , it shows that there is no significant difference between our datasets.

CONCLUSION: In this practical, we trained the given data set according to the need and test values were generated. We used the train_test_split function. We used the seaborn library to plot the histogram which helped us to confirm our splitting proportion. We used the count function to identify the records of training data. We used z_test to validate the partition. In our case null hypothesis is accepted which shows that there is no significant difference between trained and test dataset.