```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
import numpy as np
data = pd.read_csv("/content/drive/MyDrive/IBM_nalaiyathiran/Churn_Modelling.csv")
data
```

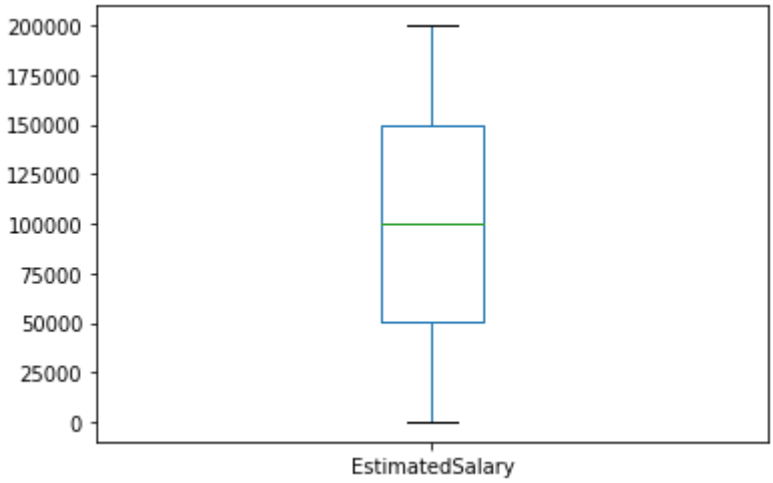
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenur
0	1	15634602	Hargrave	619	France	Female	42	
1	2	15647311	Hill	608	Spain	Female	41	
2	3	15619304	Onio	502	France	Female	42	
3	4	15701354	Boni	699	France	Female	39	
4	5	15737888	Mitchell	850	Spain	Female	43	
...	...	...	...	...	...	...	...	.
9995	9996	15606229	Obijiaku	771	France	Male	39	
9996	9997	15569892	Johnstone	516	France	Male	35	1
9997	9998	15584532	Liu	709	France	Female	36	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	
			Walker	792	France	Female	28	

Saving...

```
data.head()
```

```
data.boxplot(column =['EstimatedSalary'], grid = False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d4b043e10>
```




```
data.isnull().sum()
```

```
RowNumber          0
CustomerId         0
Surname            0
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
[illegible]        0
Exited             0
dtype: int64
```

Saving...

```
data.describe()
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Ba
count	10000.00000	1.000000e+04	10000.000000	10000.000000	10000.000000	10000.0

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['Geography']=le.fit_transform(data['Geography'])
data['Gender']=le.fit_transform(data['Gender'])
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	
0	1	15634602	Hargrave	619	0	0	42	2	
1	2	15647311	Hill	608	2	0	41	1	
2	3	15619304	Onio	502	0	0	42	8	1
3	4	15701354	Boni	699	0	0	39	1	
4	5	15737888	Mitchell	850	2	0	43	2	1

```
y=data['EstimatedSalary']
x=data.drop(columns=['EstimatedSalary','Surname'],axis=1)
```

```
names=x.columns
```

Saving...

```
d', 'CreditScore', 'Geography', 'Gender', 'Age',
       'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember',
       'Exited'],
      dtype='object')
```

```
from sklearn.preprocessing import scale
x=scale(x)
x
```

```
array([[-1.73187761, -0.78321342, -0.32622142, ...,  0.64609167,
         0.97024255,  1.97716468],
       [-1.7315312 , -0.60653412, -0.44003595, ..., -1.54776799,
         0.97024255, -0.50577476],
       [-1.73118479, -0.99588476, -1.53679418, ...,  0.64609167,
        -1.03067011,  1.97716468],
       ...,
       [ 1.73118479, -1.47928179,  0.60498839, ..., -1.54776799,
         0.97024255,  1.97716468],
       [ 1.7315312 , -0.11935577,  1.25683526, ...,  0.64609167,
        -1.03067011,  1.97716468],
       [ 1.73187761, -0.87055909,  1.46377078, ...,  0.64609167,
        -1.03067011, -0.50577476]])
```

```
x=pd.DataFrame(x,columns=names)
x.head()
```

	RowNumber	CustomerId	CreditScore	Geography	Gender	Age	Tenure	I
0	-1.731878	-0.783213	-0.326221	-0.901886	-1.095988	0.293517	-1.041760	-1
1	-1.731531	-0.606534	-0.440036	1.515067	-1.095988	0.198164	-1.387538	0
2	-1.731185	-0.995885	-1.536794	-0.901886	-1.095988	0.293517	1.032908	1
3	-1.730838	0.144767	0.501521	-0.901886	-1.095988	0.007457	-1.387538	-1
4	-1.730492	0.652659	2.063884	1.515067	-1.095988	0.388871	-1.041760	0

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
x_train
```

	RowNumber	CustomerId	CreditScore	Geography	Gender	Age	Tenure
8243	1.123581	-1.651082	1.805214	-0.901886	0.912419	-0.946079	-0.695982
9488	1.554862	-0.439100	0.211811	1.515067	0.912419	-0.660018	0.341352
6474	0.510782	-0.900810	0.460134	-0.901886	0.912419	-0.183251	-1.041760
			0.894698	1.515067	-1.095988	0.579578	-1.041760
			0.677416	-0.901886	-1.095988	0.484225	-1.387538
...	...	...	...	...	...	...	..
9298	1.489044	-0.596302	0.946432	0.306591	0.912419	-1.136786	-0.004426
6951	0.676019	0.666575	0.077303	-0.901886	0.912419	0.293517	1.032908
7291	0.793799	-1.106905	-0.533157	1.515067	0.912419	-0.564665	-0.350204
2742	-0.782021	-1.596141	-0.015818	-0.901886	-1.095988	0.198164	-1.041760
7066	0.715857	-0.143156	0.811924	1.515067	-1.095988	0.293517	-1.387538

8000 rows × 12 columns

Saving...

```
import tensorflow

from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense


model = Sequential()


model.add(Dense(units = 4,kernel_initializer= "random_uniform",activation= "relu"))


model.add(Dense(units = 10,kernel_initializer= "random_uniform",activation= "relu"))


model.add(Dense(units = 12,kernel_initializer= "random_uniform",activation= "relu"))


model.add(Dense(units = 32,kernel_initializer= "random_uniform",activation= "relu"))


model.add(Dense(units = 1 ,kernel_initializer= "random_uniform"))



model.compile(loss="mse",metrics=["mse"],optimizer="adam")


model.fit(x_train,y_train,epochs=10,batch_size=10)
```

```
Epoch 1/10
800/800 [==============================] - 2s 2ms/step - loss: 11321431040.0000 - ms
Epoch 2/10
800/800 [==============================] - 1s 2ms/step - loss: 3643340288.0000 - mse
Epoch 3/10
800/800 [==============================] - 1s 2ms/step - loss: 3409902848.0000 - mse
Epoch 4/10
                                        ========] - 1s 2ms/step - loss: 3380759552.0000 - mse
                                        ========] - 1s 2ms/step - loss: 3363616256.0000 - mse
Epoch 6/10
800/800 [==============================] - 1s 2ms/step - loss: 3355409152.0000 - mse
Epoch 7/10
800/800 [==============================] - 1s 2ms/step - loss: 3347043840.0000 - mse
Epoch 8/10
800/800 [==============================] - 1s 2ms/step - loss: 3342067456.0000 - mse
Epoch 9/10
800/800 [==============================] - 1s 2ms/step - loss: 3339269888.0000 - mse
Epoch 10/10
800/800 [==============================] - 1s 2ms/step - loss: 3334885632.0000 - mse
<keras.callbacks.History at 0x7f3cdfe28ed0>
```

Saving...

```
pred = model.predict(x_test)
pred
```

```
63/63 [==============================] - 0s 1ms/step
array([[103847.41 ],
       [ 99726.73 ],
       [ 96623.8  ],
       ...,
       [ 94247.2  ],
```

```
       [102013.734],
       [ 98707.766]], dtype=float32)
```

```
from sklearn import metrics
metrics.mean_squared_error(pred,y_test)
```

```
3259410733.7873964
```

Saving...

oducts - Cancel contracts here

ompleted at 2:09 PM