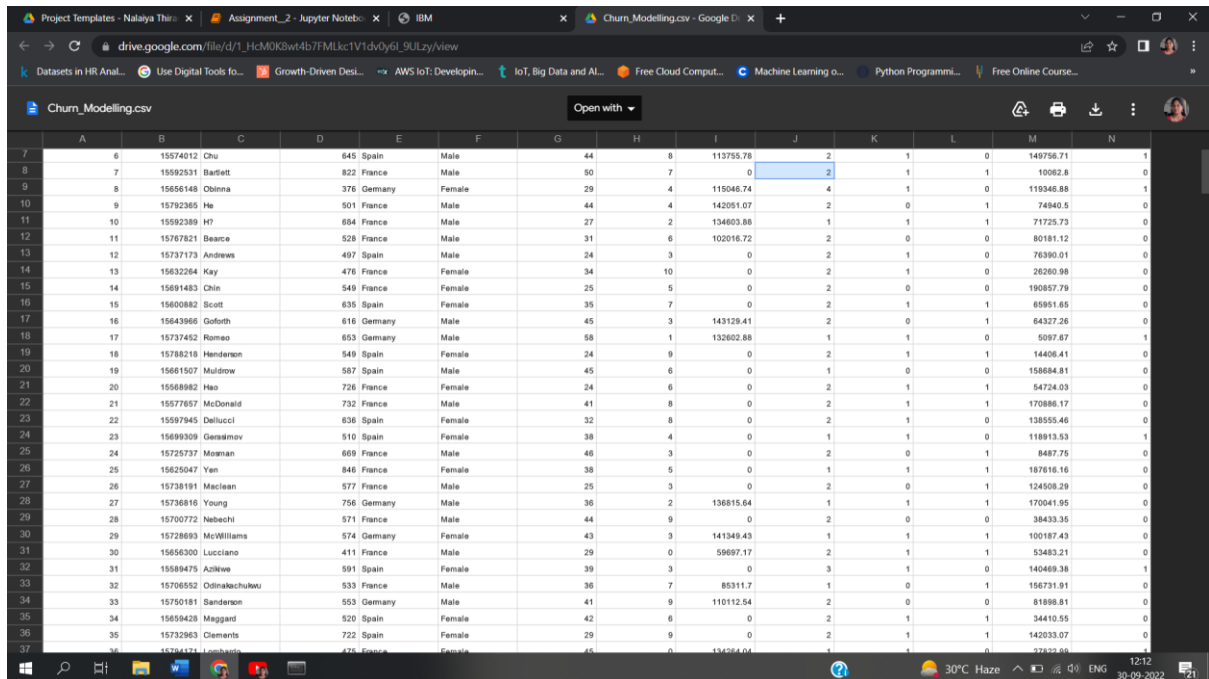


Assignment -2

Applied Data Science

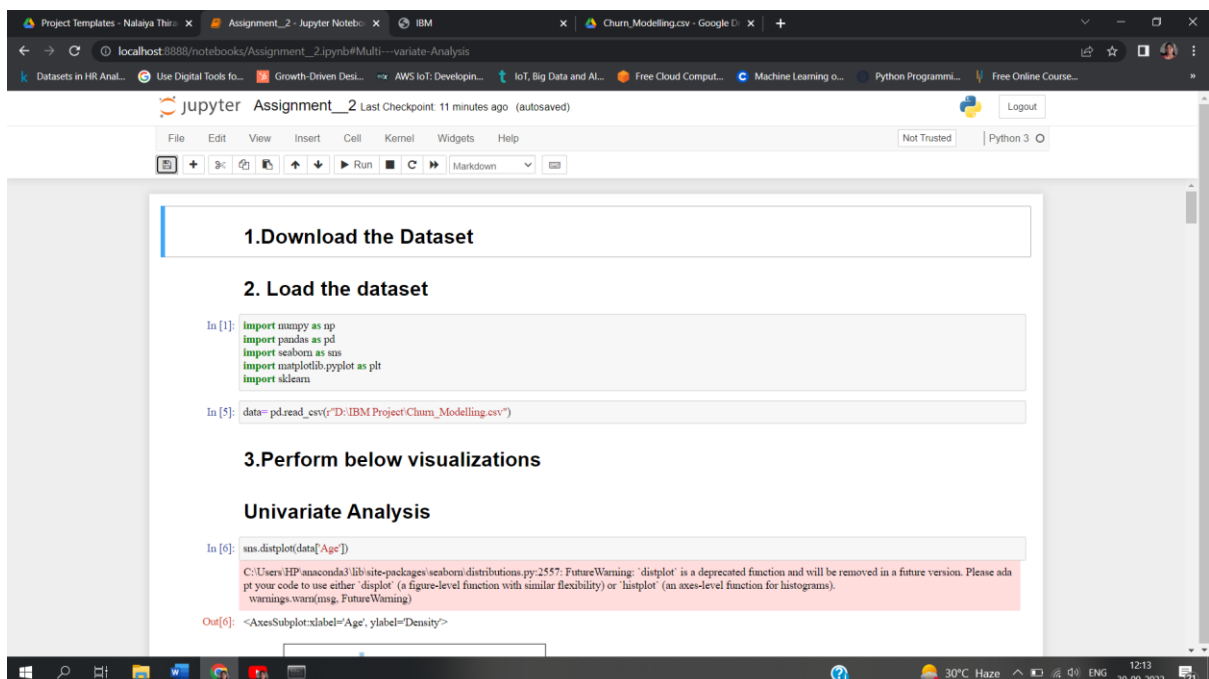
Assignment Date	30 September 2022
Student Name	Varshini P
Student Roll Number	721719104089
Maximum Marks	2 Marks

1.Download the dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
7	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
8	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
9	8	15656148	Olinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
10	9	15792385	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
11	10	15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
12	11	15767821	Beance	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
13	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390.01	0
14	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26260.98	0
15	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190857.79	0
16	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	69951.85	0
17	16	15643966	Gothof	616	Germany	Male	45	3	143129.41	2	0	1	64327.26	0
18	17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	1	0	5097.67	1
19	18	15788218	Henderson	548	Spain	Female	24	9	0	2	1	1	14406.41	0
20	19	15681507	Muldrov	587	Spain	Male	45	6	0	1	0	0	156864.81	0
21	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724.03	0
22	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886.17	0
23	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555.46	0
24	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118913.53	1
25	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0
26	25	15625047	Yen	846	France	Female	38	5	0	1	1	1	167616.16	0
27	26	15738191	Maclean	577	France	Male	25	3	0	2	0	1	124509.29	0
28	27	15736816	Young	756	Germany	Male	36	2	136815.64	1	1	1	170041.95	0
29	28	15700772	Nebuchi	571	France	Male	44	9	0	2	0	1	38433.35	0
30	29	15728693	McWilliams	574	Germany	Female	43	3	141349.43	1	1	1	100187.43	0
31	30	15656300	Lucciano	411	France	Male	29	0	59697.17	2	1	1	53463.21	0
32	31	15589475	Azilewa	591	Spain	Female	39	3	0	3	1	0	140469.38	1
33	32	15706552	Olinakochukwu	533	France	Male	36	7	85311.7	1	0	1	156731.91	0
34	33	15750181	Sanderson	553	Germany	Male	41	9	110112.54	2	0	0	81896.81	0
35	34	15659428	Maggard	520	Spain	Female	42	6	0	2	1	1	34410.55	0
36	35	15732963	Clements	722	Spain	Female	29	9	0	2	1	1	142033.07	0
37	36	15732963	Clements	722	Spain	Female	29	9	0	2	1	1	142033.07	0

2.Load the dataset



```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn

In [5]: data = pd.read_csv("D:\\IBM Project\\Churn_Modelling.csv")

3.Perform below visualizations

Univariate Analysis

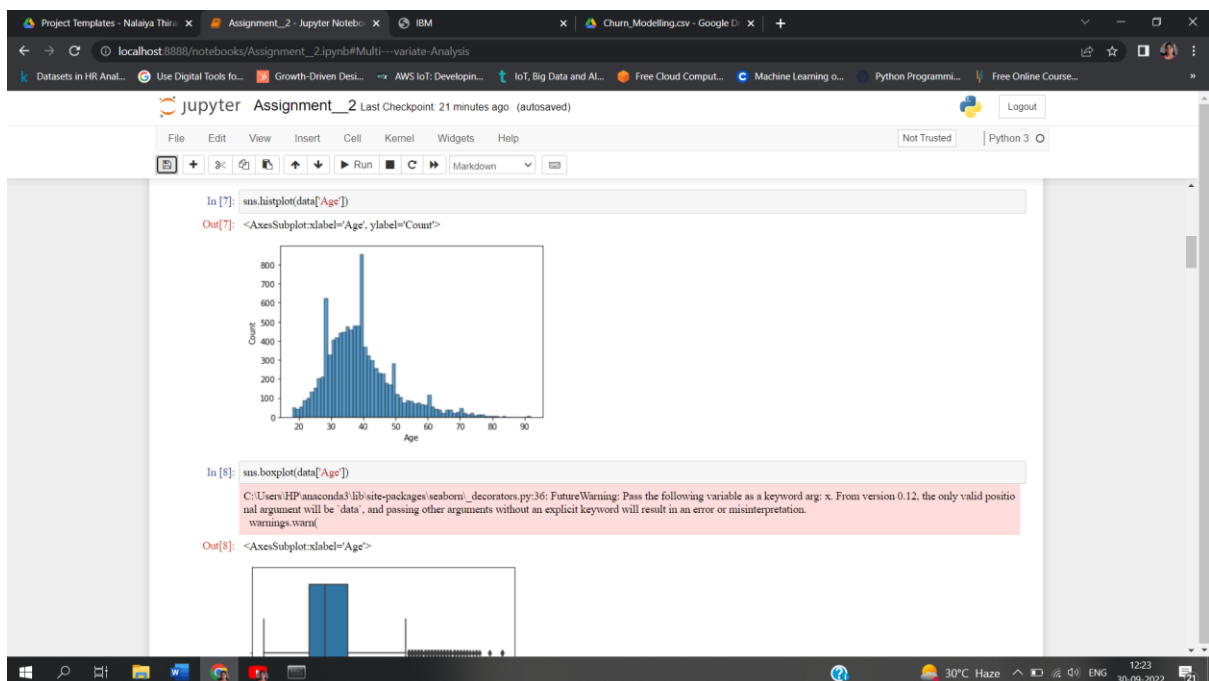
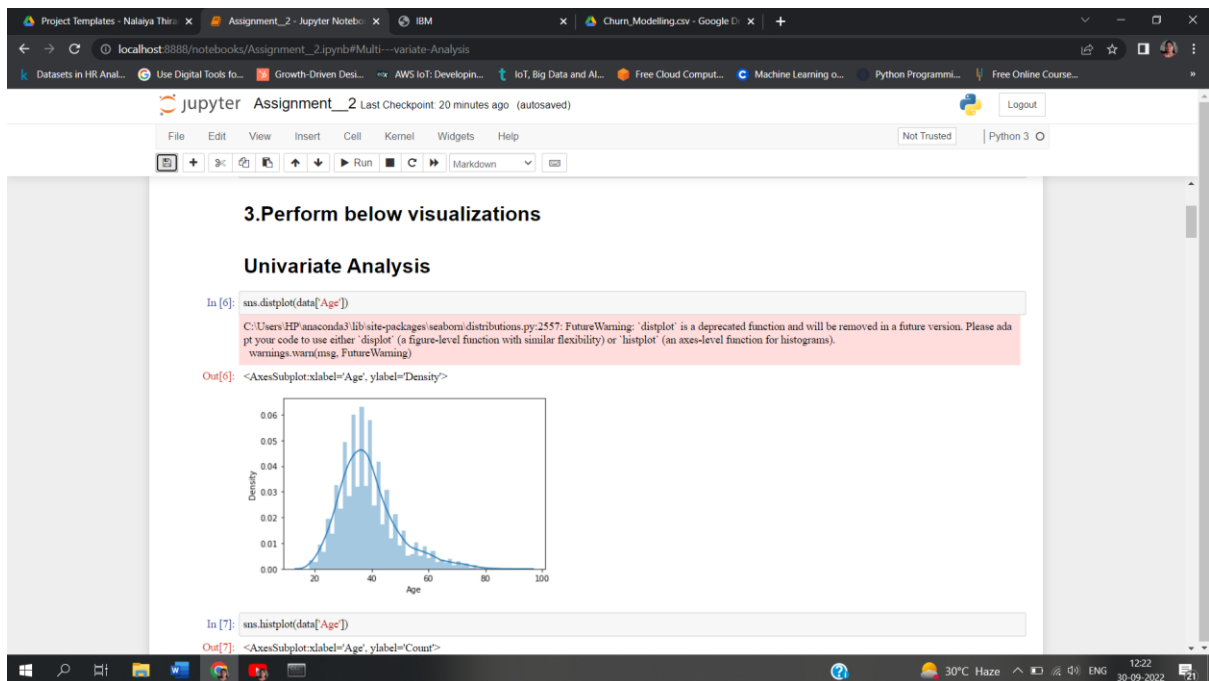
In [6]: sns.distplot(data['Age'])

C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

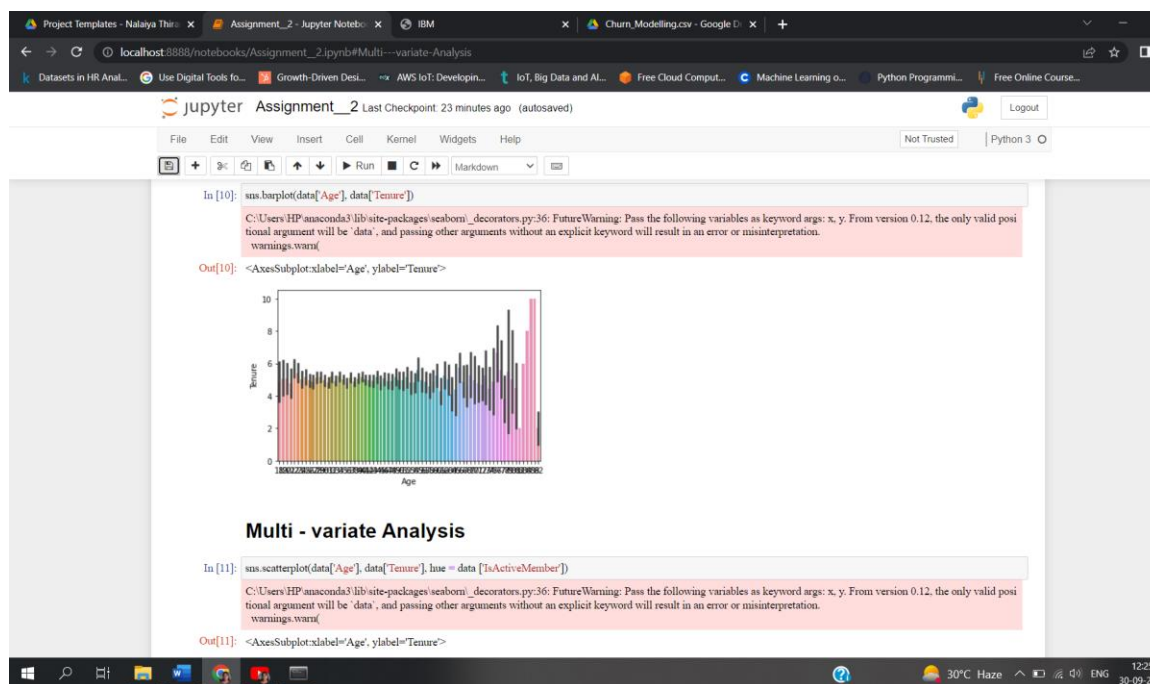
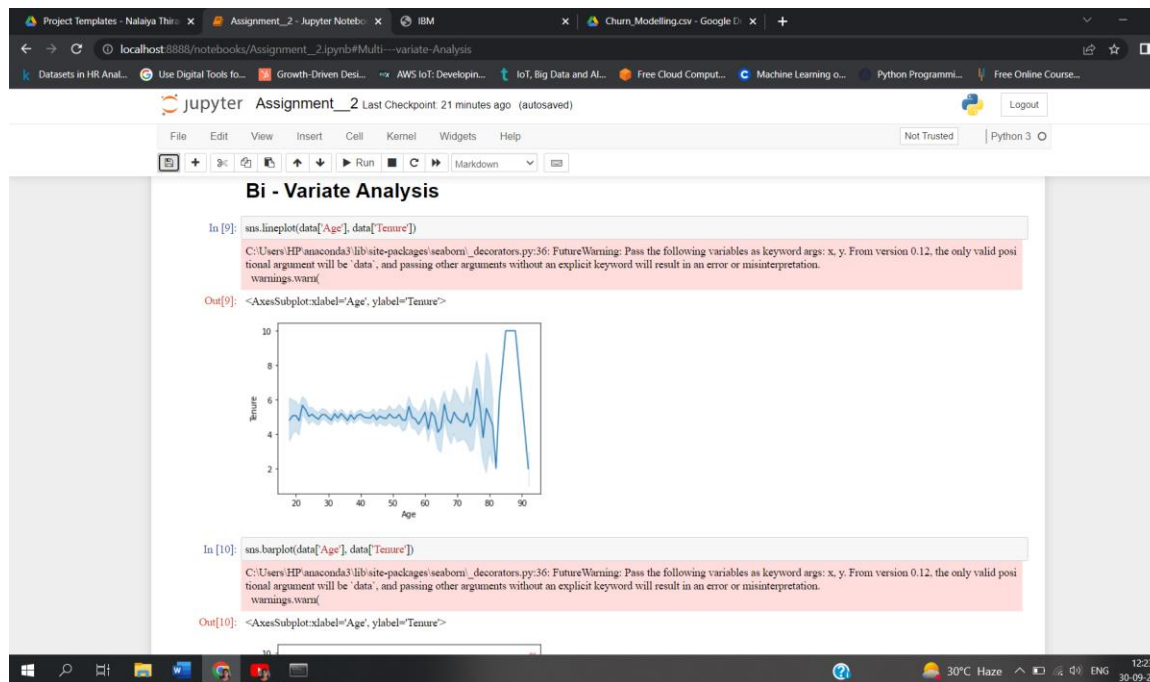
Out[6]: <AxesSubplot:label='Age', ylabel='Density'>
```

3.Perform below visualizations

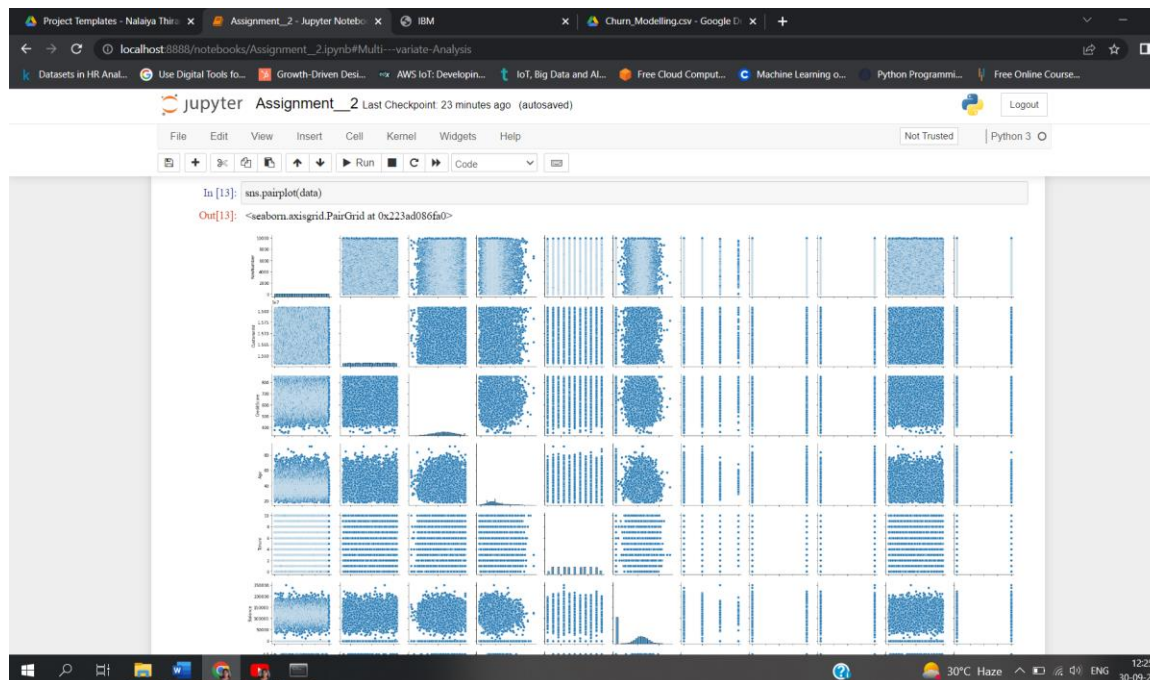
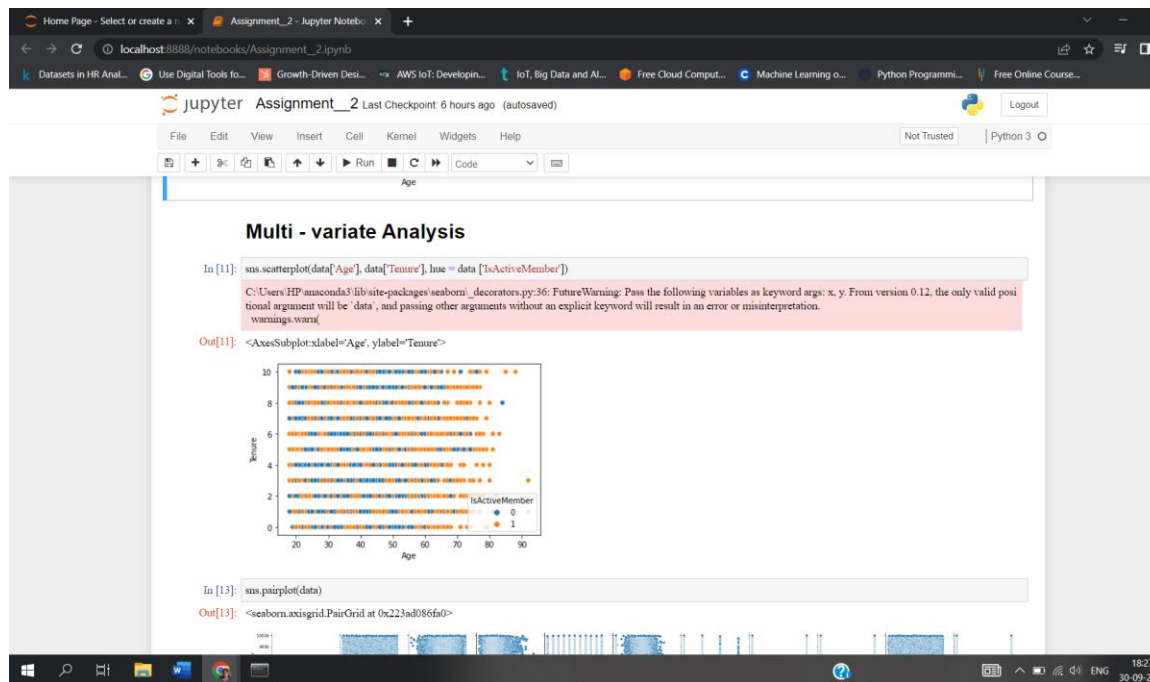
Univariate Analysis

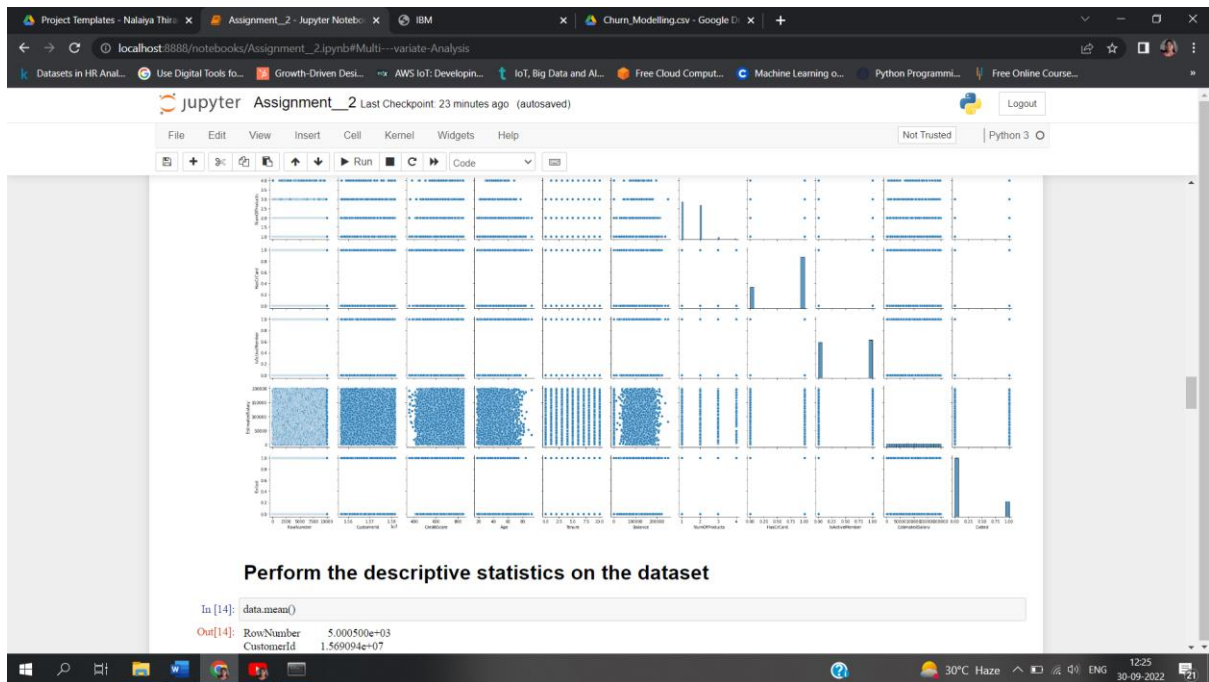


Bi Variate Analysis

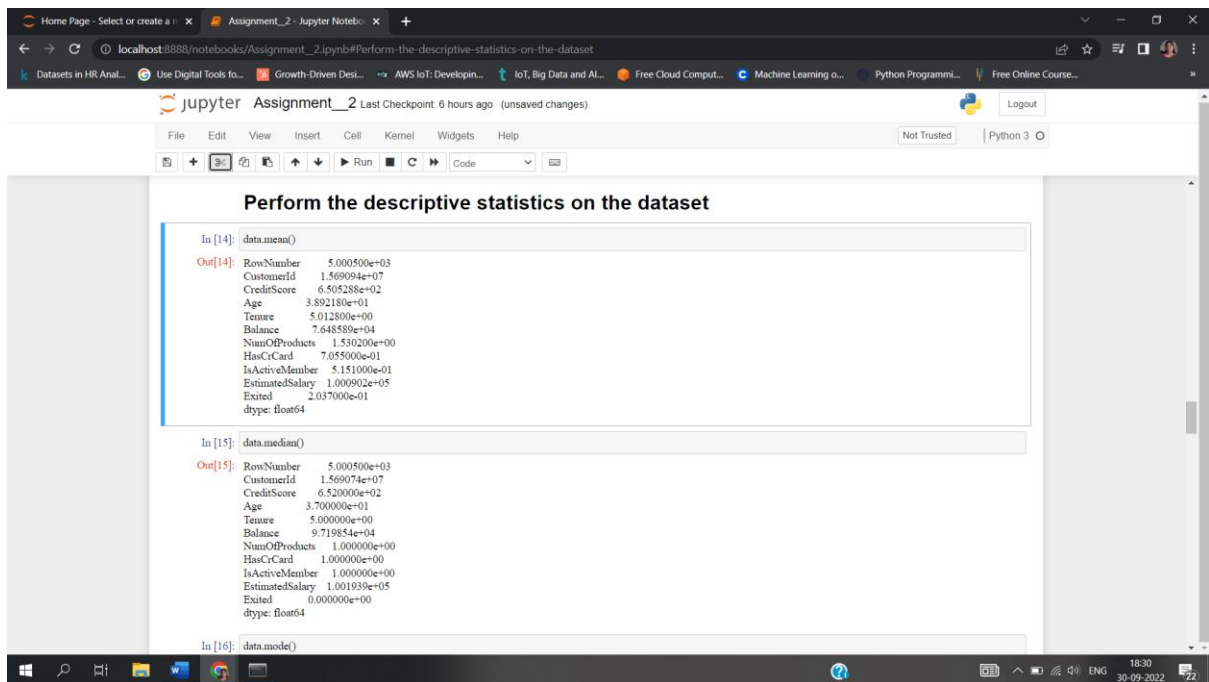


Multi Variate Analysis





4. Perform the descriptive statistics on the dataset



Home Page - Select or create a notebook | Assignment_2 - Jupyter Notebook | +

localhost:8888/notebooks/Assignment_2.ipynb#Perform-the-descriptive-statistics-on-the-dataset

Jupyter Assignment_2 Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```
Age 3.700000e+01
Tenure 5.000000e+00
Balance 9.719854e+04
NumOfProducts 1.000000e+00
HasCrCard 1.000000e+00
IsActiveMember 1.000000e+00
EstimatedSalary 1.001939e+05
Exited 0.000000e+00
dtype: float64
```

In [16]: data.mode()

Out[16]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.5
1	2	15565706	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	5	15565796	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

10000 rows x 14 columns

5. Handling the missing values

Home Page - Select or create a notebook | Assignment_2 - Jupyter Notebook | +

localhost:8888/notebooks/Assignment_2.ipynb#Perform-the-descriptive-statistics-on-the-dataset

Jupyter Assignment_2 Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Handle the missing values

```
In [17]: data.isnull().any()
```

Out[17]:

```
RowNumber      False
CustomerId      False
Surname         False
CreditScore     False
Geography       False
Gender          False
Age             False
Tenure          False
Balance         False
NumOfProducts   False
HasCrCard       False
IsActiveMember  False
EstimatedSalary False
Exited          False
dtype: bool
```

```
In [18]: data.isnull().sum()
```

Out[18]:

```
RowNumber      0
CustomerId      0
Surname         0
CreditScore     0
Geography       0
Gender          0
Age             0
Tenure          0
Balance         0
NumOfProducts   0
HasCrCard       0
IsActiveMember  0
```


6. Find and replace the outliers.

The screenshot shows a Jupyter Notebook interface with the following content:

```
HasCrCard    0
IsActiveMember 0
EstimatedSalary
Exited      0
dtype: int64
```

There are no missing values

6. Find the outliers and replace the outliers

In [19]: `data.quantile([0.1])`

Out[19]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.1	1000.9	15591167.1	521.0	27.0	1.0	0.0	1.0	0.0	0.0	20273.58	0.0

In [20]: `data.quantile([0.1,0.9])`

Out[20]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.1	1000.9	15591167.1	521.0	27.0	1.0	0.000	1.0	0.0	0.0	20273.580	0.0
0.9	9000.1	15790830.7	778.0	53.0	9.0	149244.792	2.0	1.0	1.0	179674.704	1.0

In [21]: `data.quantile([0.1,0.5])`

Out[21]:

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0.1	1000.9	15591167.1	521.0	27.0	1.0	0.00	1.0	0.0	0.0	20273.580	0.0
0.5	5000.5	15690738.0	652.0	37.0	5.0	97198.54	1.0	1.0	1.0	100193.915	0.0

7. Check for the categorical columns and perform encoding.

The screenshot shows a Jupyter Notebook interface with the following content:

```
In [22]: from sklearn import preprocessing
In [24]: le=preprocessing.LabelEncoder()
In [27]: onehot = preprocessing.OneHotEncoder()
In [28]: data['Age'] = le.fit_transform(data['Age'])
In [29]: data.head()
```

Out[29]:

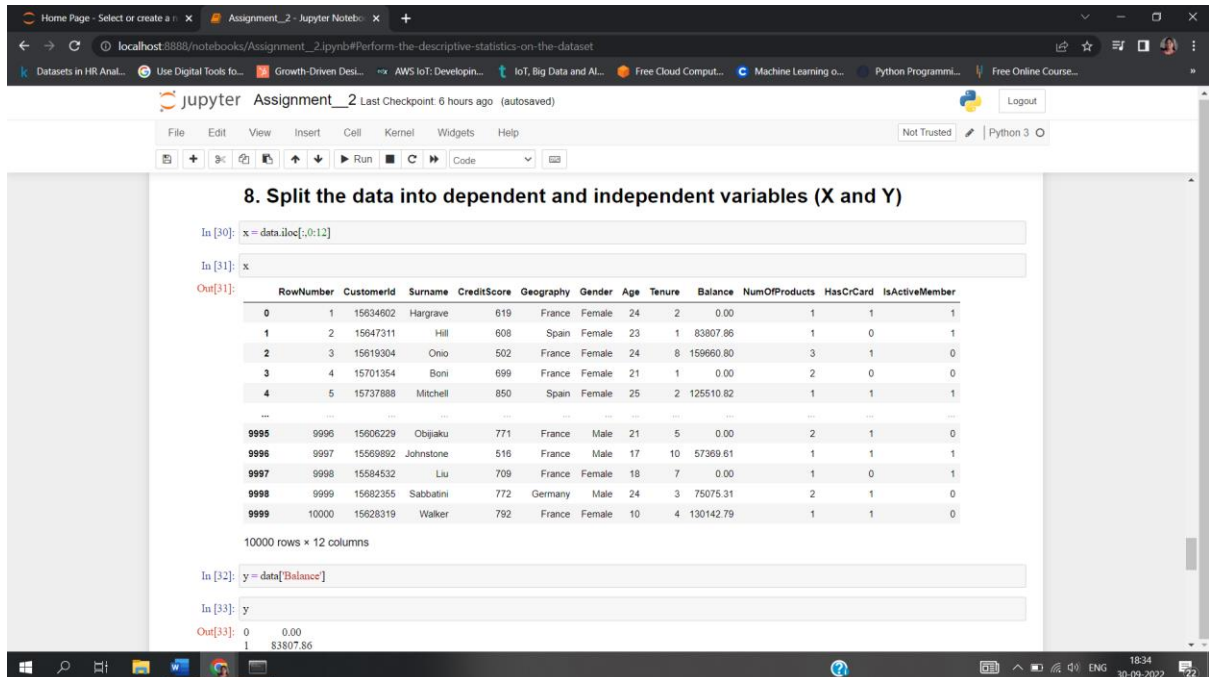
	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0	1	15634602	Hargrave	619	France	Female	24	2	0.00	1	1	1	101348.88
1	2	15647311	Hill	608	Spain	Female	23	1	83807.86	1	0	1	112542.58
2	3	15619304	Onio	502	France	Female	24	8	159960.80	3	1	0	113931.57
3	4	15701354	Boni	699	France	Female	21	1	0.00	2	0	0	93829.63
4	5	15737888	Mitchell	850	Spain	Female	25	2	125510.82	1	1	1	79084.10

8. Split the data into dependent and independent variables (X and Y)

In [30]: `x = data.iloc[:,0:12]`

In [31]: `x`

8. Split the data into dependent and independent variables (X and Y).



The screenshot shows a Jupyter Notebook interface with the title "Assignment_2". The notebook contains the following code and output:

```
In [30]: x = data.iloc[:,0:12]
```

```
In [31]: x
```

Out[31]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hairgrave	619	France	Female	24	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	23	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	24	8	159660.80	3	1	0
3	4	15701354	Boni	699	France	Female	21	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	25	2	125510.82	1	1	1
...
9995	9996	15606229	Objaku	771	France	Male	21	5	0.00	2	1	0
9996	9997	15606892	Johnstone	516	France	Male	17	10	57369.61	1	1	1
9997	9998	15584532	Liu	709	France	Female	18	7	0.00	1	0	1
9998	9999	15682355	Sabbatini	772	Germany	Male	24	3	75075.31	2	1	0
9999	10000	15628319	Walker	792	France	Female	10	4	130142.79	1	1	0

10000 rows x 12 columns

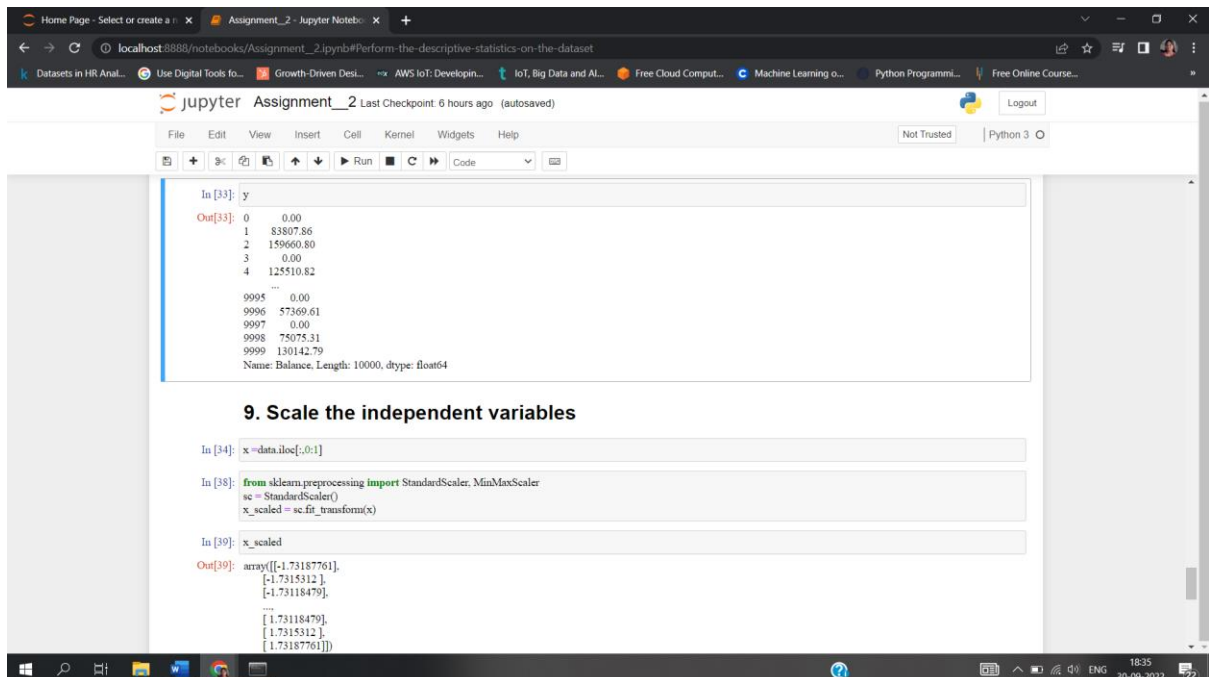
```
In [32]: y = data['Balance']
```

```
In [33]: y
```

Out[33]:

```
0    0.00
1   83807.86
2  159660.80
3    0.00
4  125510.82
...
```

9. Scale independent variables.



The screenshot shows a Jupyter Notebook interface with the title "Assignment_2". The notebook contains the following code and output:

```
In [33]: y
```

Out[33]:

```
0    0.00
1   83807.86
2  159660.80
3    0.00
4  125510.82
...
```

9. Scale the independent variables

```
In [34]: x = data.iloc[:,0:1]
```

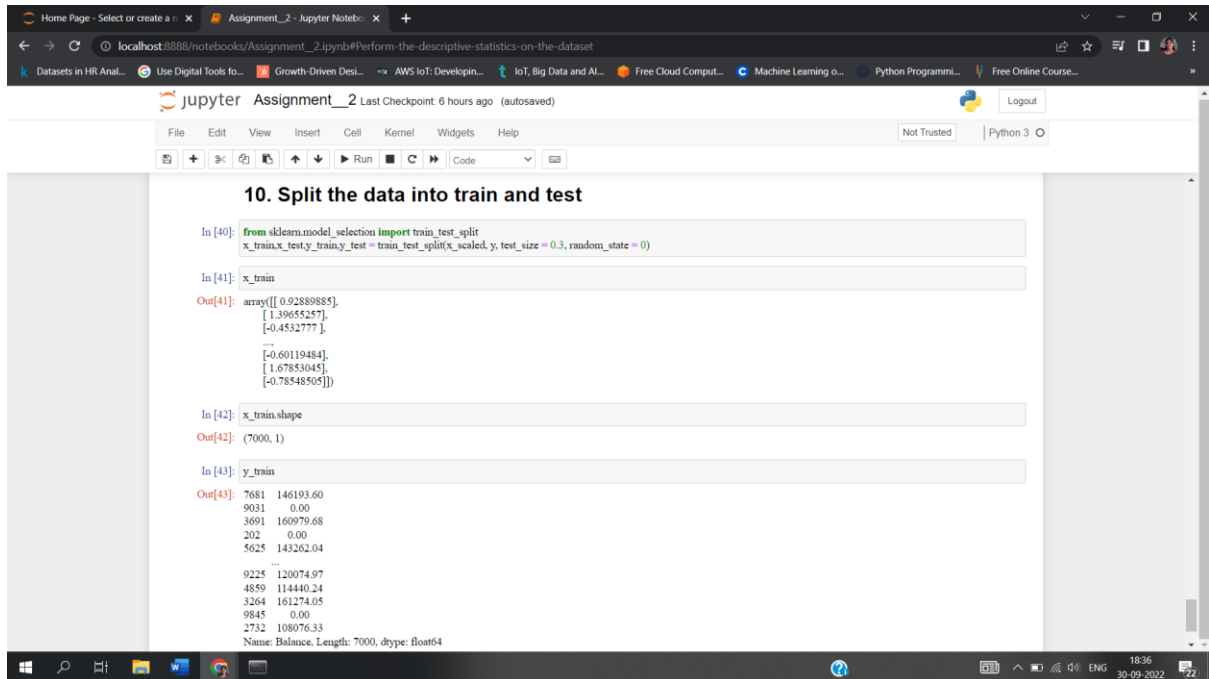
```
In [38]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_scaled = sc.fit_transform(x)
```

```
In [39]: x_scaled
```

Out[39]:

```
array([[ -1.73187761],
       [ -1.7315312 ],
       [ -1.73118479],
       [  1.73118479],
       [  1.7315312 ]])
```


10. Split the data into test and train.



The screenshot shows a Jupyter Notebook interface with the title "Assignment__2" and a "Last Checkpoint: 6 hours ago (autosaved)" status. The notebook is running on a local host at 8888. The code in the notebook is as follows:

```
In [40]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y, test_size = 0.3, random_state = 0)
```

The output for cell 41 is:

```
Out[41]: array([[ 0.92889885],
 [ 1.39655257],
 [-0.4532777 ]],
 ...,
 [-0.60119484],
 [ 1.67853045],
 [-0.78548505]])
```

The code in cell 42 is:

```
In [42]: x_train.shape
```

The output for cell 42 is:

```
Out[42]: (7000, 1)
```

The code in cell 43 is:

```
In [43]: y_train
```

The output for cell 43 is:

```
Out[43]: 7681 146193.60
9031 0.00
3691 160979.68
202 0.00
5625 143262.04
...
9225 120074.97
4859 114440.24
3264 161274.05
9845 0.00
2732 108076.33
Name: Balance, Length: 7000, dtype: float64
```

The notebook interface includes a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The status bar at the bottom shows the Windows taskbar with the time 18:36 and date 30-09-2022.