ASSIGNMENT 2

Python programming

Assignment date:26.09.2022

Student name: M. AARTHI

Student Roll Number:923819106001

Maximum Marks:2 Mark

import pandas as pd import seaborn as sns import numpy as np from matplotlib import pyplot as plt

%matplotlib inline df = pd.read\_csv("/content/Churn\_Modelling.csv")

df

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenur

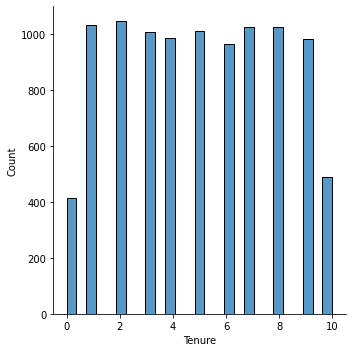
1. 1 15634602 Hargrave 619 France Female 42 2
2. 2 15647311 Hill 608 Spain Female 41
3. 3 15619304 Onio 502 France Female 42 8
4. 4 15701354 Boni 699 France Female 39 4 5 15737888 Mitchell 850 Spain Female 43 2

... ... ... ... ... ... ... ... .

1. 9996 15606229 Obijiaku 771 France Male 39 5
2. 9997 15569892 Johnstone 516 France Male 35 10
3. 9998 15584532 Liu 709 France Female 36 7
4. 9999 15682355 Sabbatini 772 Germany Male 42 3
5. 10000 15628319 Walker 792 France Female 28 4
6. rows × 14 columns

sns.displot(df.Tenure)

<seaborn.axisgrid.FacetGrid at 0x7f906f885510>

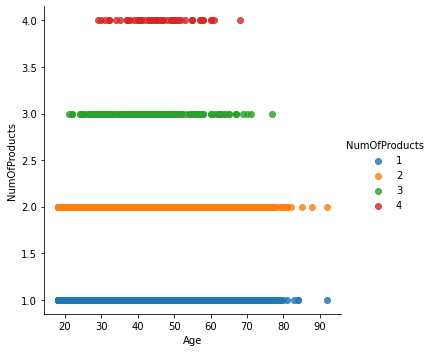


df.plot.line()

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f905a6079d0>



sns.lmplot("Age","NumOfProducts",df,hue="NumOfProducts", fit\_reg=False); /usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: P FutureWarning



RowNumber

CustomerId

CreditScore

Age

Tenure

Ba

count

10000.00000

1.000000e+04

10000.000000

10000.000000

10000.000000

10000.0

mean

5000.50000

1.569094e+07

650.528800

38.921800

5.012800

76485.8

std

2886.89568

7.193619e+04

96.653299

10.487806

2.892174

62397.4

min

1.00000

1.556570e+07

350.000000

18.000000

0.000000

0.0

25

%

2500.75000

1.562853e+07

584.000000

32.000000

3.000000

0.0

%

50

5000.50000

1.569074e+07

652.000000

37.000000

5.000000

97198.5

75

%

7500.25000

1.575323e+07

718.000000

44.000000

7.000000

127644.2

max

10000.00000

1.581569e+07

850.000000

92.000000

10.000000

250898.0

df.describe()

data = pd.read\_csv(

"Churn\_Modelling.csv"

)

pd.isnull(data["Gender"])

1. False
2. False
3. False
4. False
5. False

...

1. False
2. False
3. False
4. False
5. False

Name: Gender, Length: 10000, dtype: bool

df["Tenure"] = np.where(df["Tenure"] >10, np.median,df["Tenure"]) df["Tenure"]

1. 2
2. 1
3. 8
4. 1
5. 2 ..
6. 5
7. 10
8. 7
9. 3
10. 4

Name: Tenure, Length: 10000, dtype: object

pd.get\_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"]) df.head()

RowNumber CustomerId Surname CreditScore Geography Gender Age Tenure

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0.0000 | 15634602 | Hargrave | 619 | France | Female | 42 | 2 |  |
| 1 | 0.0001 | 15647311 | Hill | 608 | Spain | Female | 41 | 1 | 8 |
| 2 | 0.0002 | 15619304 | Onio | 502 | France | Female | 42 | 8 | 15 |
| 3 | 0.0003 | 15701354 | Boni | 699 | France | Female | 39 | 1 |  |
| 4 | 0.0004 | 15737888 | Mitchell | 850 | Spain | Female | 43 | 2 | 12 |

1. = df.iloc[:, :-2].values print(X)

[[1 15634602 'Hargrave' ... 1 1 1]

* 1. 15647311 'Hill' ... 1 0 1]
  2. 15619304 'Onio' ... 3 1 0]

...

* 1. 15584532 'Liu' ... 1 0 1]
  2. 15682355 'Sabbatini' ... 2 1 0] [10000 15628319 'Walker' ... 1 1 0]]

1. = df.iloc[:, -1].values print(Y)

[1 0 1 ... 1 1 0]

import pandas as pd from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler() df[["RowNumber"]] = scaler.fit\_transform(df[["RowNumber"]]) print(df)

RowNumber CustomerId Surname CreditScore Geography Gender Age \

* 1. 0.0000 15634602 Hargrave 619 France Female 42
  2. 0.0001 15647311 Hill 608 Spain Female 41
  3. 0.0002 15619304 Onio 502 France Female 42
  4. 0.0003 15701354 Boni 699 France Female 39
  5. 0.0004 15737888 Mitchell 850 Spain Female 43 ... ... ... ... ... ... ... ...
  6. 0.9996 15606229 Obijiaku 771 France Male 39
  7. 0.9997 15569892 Johnstone 516 France Male 35
  8. 0.9998 15584532 Liu 709 France Female 36
  9. 0.9999 15682355 Sabbatini 772 Germany Male 42
  10. 1.0000 15628319 Walker 792 France Female 28

Tenure Balance NumOfProducts HasCrCard IsActiveMember \

* 1. 2 0.00 1 1 1
  2. 1 83807.86 1 0 1
  3. 8 159660.80 3 1 0
  4. 1 0.00 2 0 0
  5. 2 125510.82 1 1 1 ... ... ... ... ... ...
  6. 5 0.00 2 1 0
  7. 10 57369.61 1 1 1
  8. 7 0.00 1 0 1
  9. 3 75075.31 2 1 0
  10. 4 130142.79 1 1 0

EstimatedSalary Exited

* 1. 101348.88 1
  2. 112542.58 0
  3. 113931.57 1
  4. 93826.63 0
  5. 79084.10 0 ... ... ...
  6. 96270.64 0
  7. 101699.77 0
  8. 42085.58 1
  9. 92888.52 1
  10. 38190.78 0
  11. rows x 14 columns]

from sklearn.model\_selection import train\_test\_split train\_size=0.8

X = df.drop(columns = ['Tenure']).copy() y = df['Tenure']

X\_train, X\_rem, y\_train, y\_rem = train\_test\_split(X,y, train\_size=0.8) test\_size = 0.5

X\_valid, X\_test, y\_valid, y\_test = train\_test\_split(X\_rem,y\_rem, test\_size=0.5) print(X\_train.shape), print(y\_train.shape) print(X\_valid.shape), print(y\_valid.shape) print(X\_test.shape), print(y\_test.shape)

(8000, 13) (8000,)

(1000, 13) (1000,)

(1000, 13)

(1000,)

(None, None)

[Colab paid products](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) - [Cancel contracts here](https://colab.research.google.com/cancel-subscription)