MAHENDRAENGINEERINGCOLLEGEFORWOMEN NAME: M.Inbanila

CLASS:IVYEAR-CSE SUB:IBM(AI)

REGNO:611419104017

#libraries

importpandasaspd importnumpyasnpp

importmatplotlib.pyplotasplt

%matplotlibinline #loaddataset

df=pd.read\_csv(r"/content/Churn\_Modelling.csv")df.head(10)

RowNumberCustomerIdSurnameCreditScoreGeographyGenderAge

\

1. 115634602Hargrave 619FranceFemale42
2. 215647311Hill 608SpainFemale41
3. 315619304Onio 502FranceFemale42
4. 415701354Boni 699FranceFemale39
5. 515737888 Mitchell 850SpainFemale435 615574012 Chu 645SpainMale44
6. 715592531Bartlett 822FranceMale50
7. 815656148Obinna 376 GermanyFemale298 915792365 He 501FranceMale44

9 1015592389 H? 684FranceMale27

TenureBalanceNumOfProductsHasCrCardIsActiveMember\ 0 2 0.00 1 1 1

1 183807.86 1 0 1

2 8159660.80 3 1 0

3 1 0.00 2 0 0

4 2125510.82 1 1 1

5 8113755.78 2 1 0

6 7 0.00 2 1 1

7 4115046.74 4 1 0

8 4142051.07 2 0 1

9 2134603.88 1 1 1

EstimatedSalaryExited 0 101348.88 1

1 112542.58 0

2 113931.57 1

3 93826.63 0

4 79084.10 0

5 149756.71 1

6 10062.80 0

7 119346.88 1

8 74940.50 0

9 71725.73 0

df.info()

<class'pandas.core.frame.DataFrame'>RangeIndex:10000entries,0to9999 Datacolumns(total14columns):

#Column Non-NullCountDtype

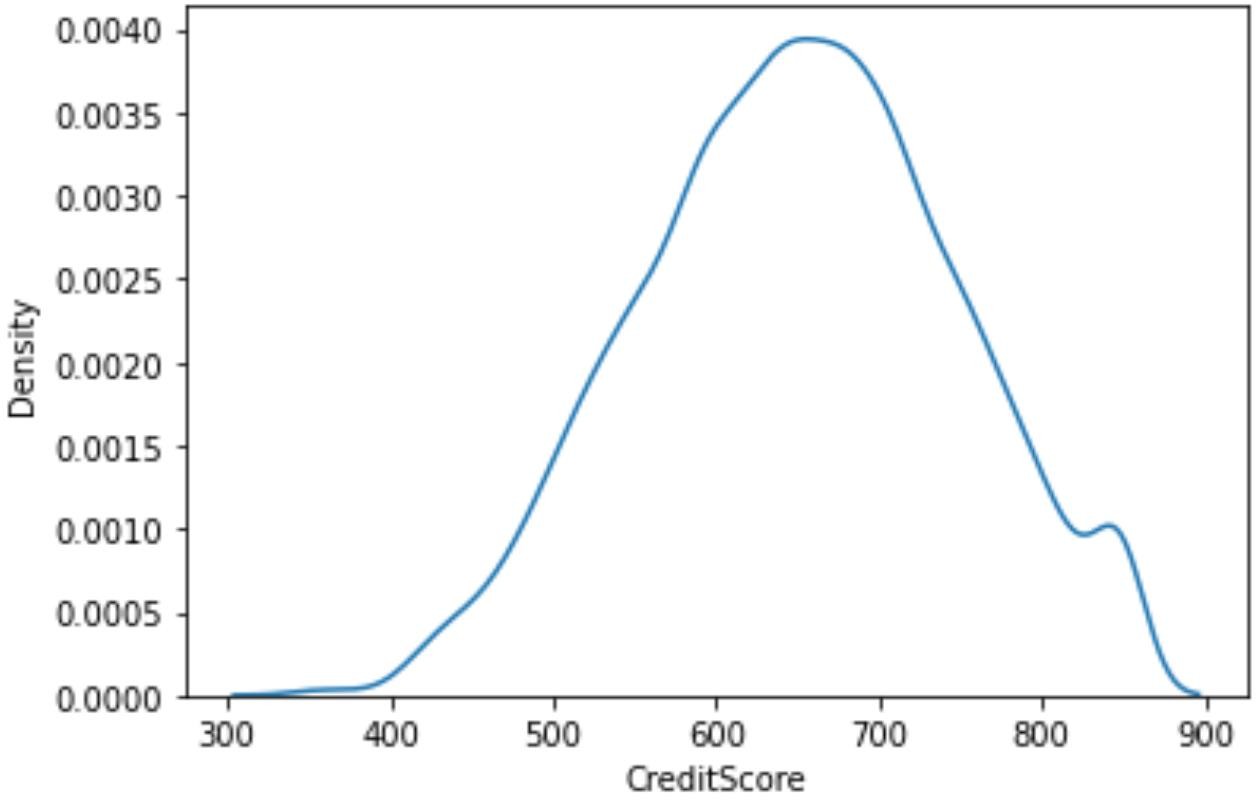
1. RowNumber 10000non-nullint64
2. CustomerId 10000non-nullint64
3. Surname 10000non-nullobject
4. CreditScore10000non-nullint64
5. Geography 10000non-nullobject
6. Gender 10000non-nullobject
7. Age 10000non-nullint64
8. Tenure 10000non-nullint64
9. Balance 10000non-nullfloat64
10. NumOfProducts10000non-nullint64 10 HasCrCard 10000non-null int64 11IsActiveMember10000non-nullint64
11. EstimatedSalary10000non-nullfloat64
12. Exited 10000non-nullint64 dtypes: float64(2),int64(9),object(3)memoryusage:1.1+MB #Visualizations

#UnivariateAnalysisimportseabornassns

sns.kdeplot(df['CreditScore'])

<matplotlib.axes.\_subplots.AxesSubplotat0x7fc4a0cd2790>

#Bi-VariateAnalysisplt.bar(df.CustomerId,df.CreditScore)plt.title('CreditScore')plt.xlabel('CustomerId')plt.ylabel('CreditScore')

Text(0,0.5,'CreditScore')

sns.lmplot(x='Tenure',y='Balance',data=df,hue='Exited',size=8)

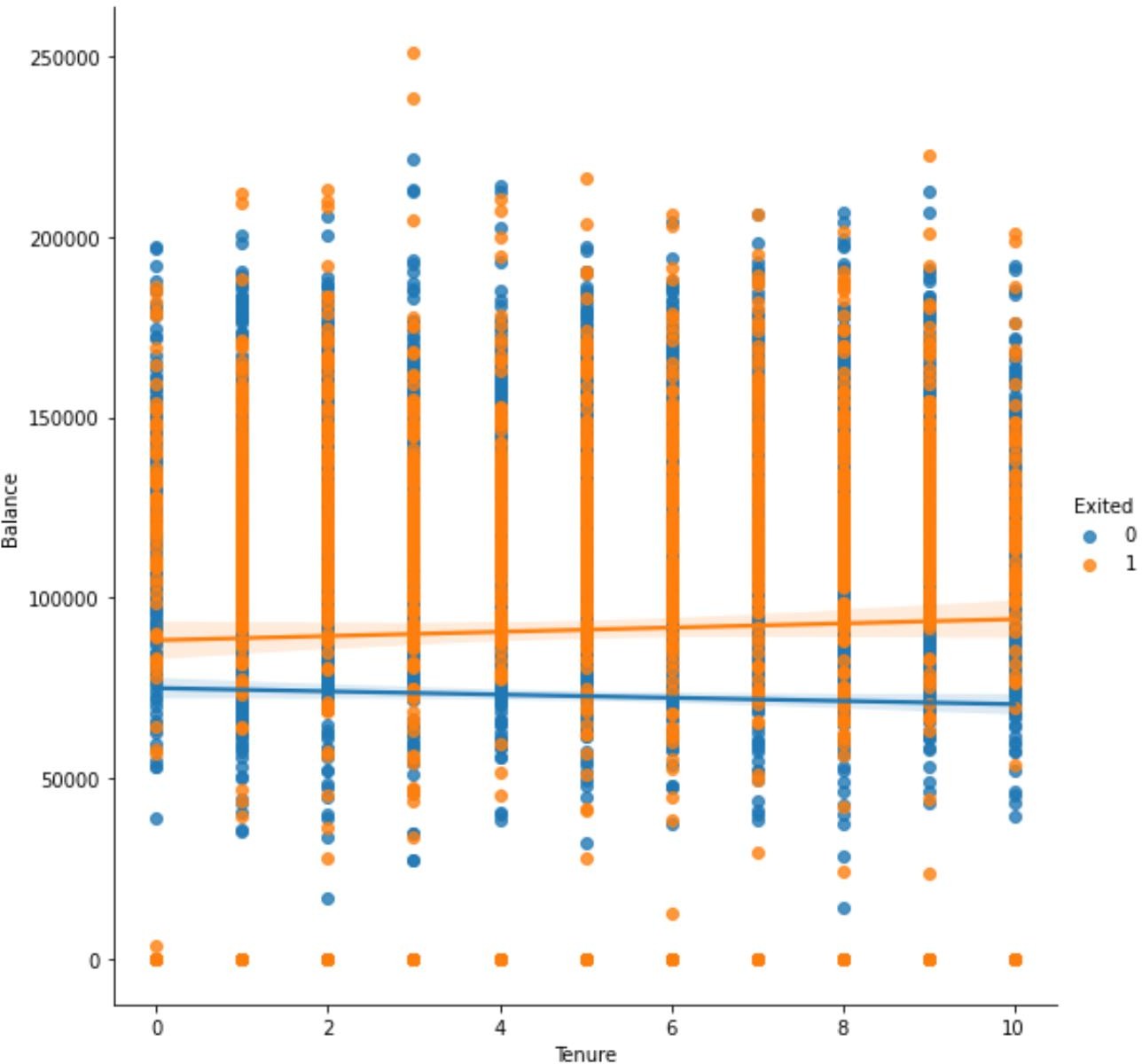
/usr/local/lib/python3.7/dist-packages/seaborn/regression.py:581:UserWarning:The`size`parameterhasbeenrenamedto`height`;pleaseupdateyourcode.

warnings.warn(msg,UserWarning)

<seaborn.axisgrid.FacetGridat0x7fc4a149e2d0>

#Multi-VariateAnalysisax=

df[["CreditScore","Age","Tenure","Balance"]].plot(figsize=(80,40))ax.legend(loc='centerleft',bbox\_to\_anchor=(1,0.5));



df.isnull().sum()RowNumber 0

CustomerId 0

Surname 0

CreditScore 0

Geography 0

Gender 0

Age 0

Tenure 0

Balance 0

NumOfProducts 0

HasCrCard 0

IsActiveMember 0

EstimatedSalary 0

Exited 0

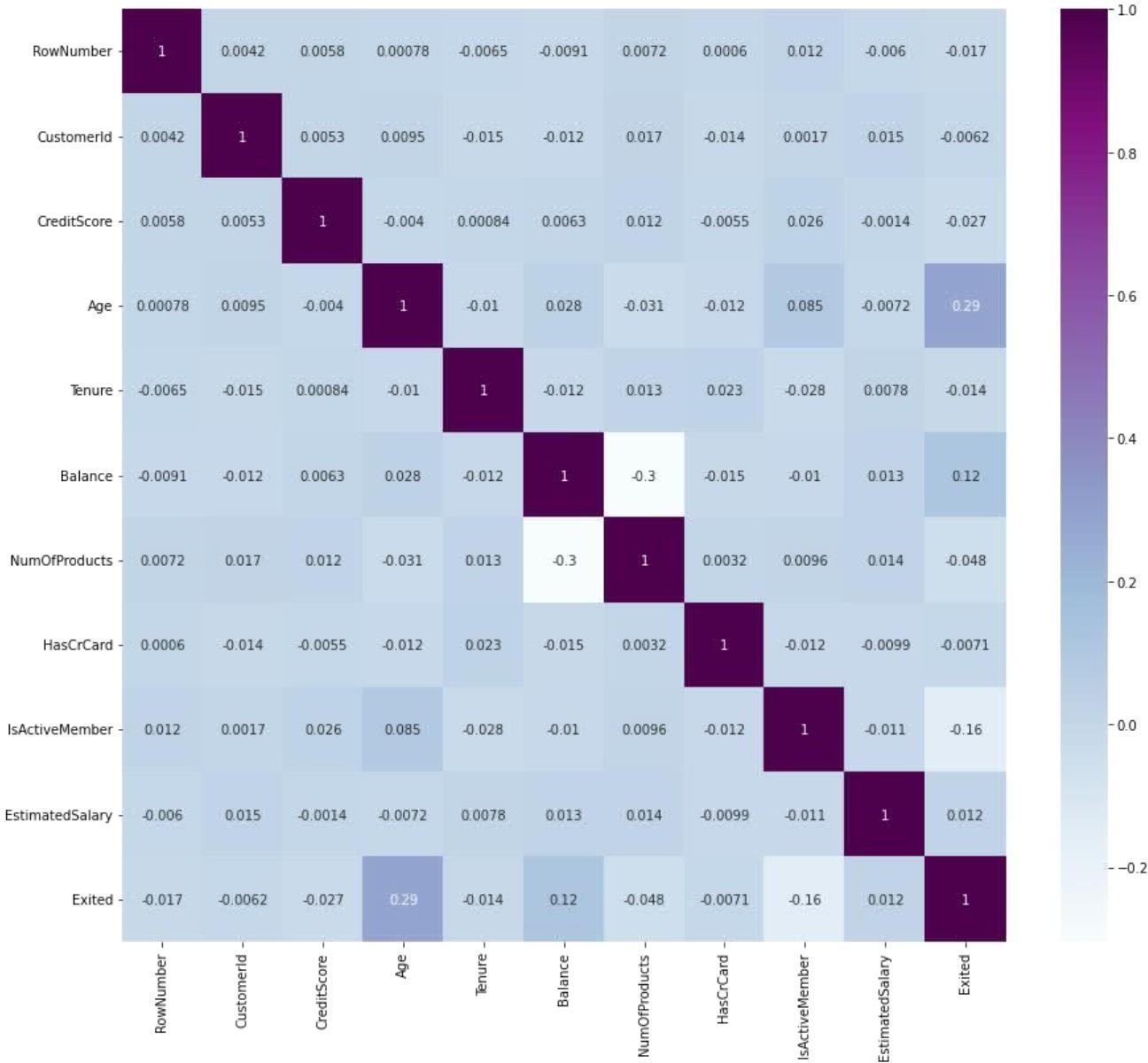
dtype:int64plt.figure(figsize=(15,13))

sns.heatmap(df.corr(),annot=True,cmap='BuPu')plt.show()

df.drop(['RowNumber','CustomerId','Surname'],axis=1,inplace=True)

df.head()

CreditScoreGeographyGender Age TenureBalanceNumOfProducts\

0 619France Female42 2 0.001

1 608SpainFemale41 183807.86

1

2 502FranceFemale42 8159660.80

3

3 699France Female39 1 0.002

4 850SpainFemale43 2125510.82

1

HasCrCardIsActiveMemberEstimatedSalaryExited 0 1 1 101348.88 1

1 0 1 112542.58 0

2 1 0 113931.57 1

3 0 0 93826.63 0

4 1 1 79084.10 0

df.info()

<class'pandas.core.frame.DataFrame'>RangeIndex:10000entries,0to9999 Datacolumns(total11columns):

#Column Non-NullCountDtype

1. CreditScore10000non-nullint64
2. Geography 10000non-nullobject
3. Gender 10000non-nullobject
4. Age 10000non-nullint64
5. Tenure 10000non-nullint64
6. Balance 10000non-nullfloat64
7. NumOfProducts10000non-nullint64
8. HasCrCard 10000non-nullint64
9. IsActiveMember10000non-nullint64
10. EstimatedSalary10000non-nullfloat64 10 Exited 10000non-null int64 dtypes:float64(2),int64(7),object(2)memoryusage:859.5+KB df["Geography"].unique()

array(['France','Spain','Germany'],dtype=object)

df["Gender"].unique()array(['Female','Male'],dtype=object)

geo=pd.get\_dummies(df["Geography"],drop\_first=False)geo.head()

FranceGermanySpain

0 1 0 0

1 0 0 1

2 1 0 0

3 1 0 0

4 0 0 1

gen=pd.get\_dummies(df["Gender"],drop\_first=False)df=pd.concat([df,geo,gen],axis=1)

df

CreditScoreGeographyGender Age TenureBalanceNumOfProducts\

0 619France Female42 2 0.001

1 608SpainFemale41 183807.86

1

2 502FranceFemale42 8159660.80

3

3 699France Female39 1 0.002

4 850SpainFemale43 2125510.82

1

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

...

9995 771FranceMale39 5 0.00

2

9996 516FranceMale351057369.61

1

9997 709FranceFemale36 7 0.00

1

9998 772GermanyMale42 375075.31

2

9999 792FranceFemale28 4130142.79

1

HasCrCardIsActiveMemberEstimatedSalaryExited FranceGermany\

0 1 1 101348.88 1 1

0

1 0 1 112542.58 0 0

0

2 1 0 113931.57 1 1

0

3 0 0 93826.63 0 1

0

4 1 1 79084.10 0 0

0

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

...

9995 1 0 96270.64 0 1

0

9996 1 1 101699.77 0 1

0

9997 0 1 42085.58 1 1

0

9998 1 0 92888.52 1 0

1

9999 1 0 38190.78 0 1

0

SpainFemaleMale0 0 10

1 1 10

2 0 10

3 0 10

4 1 10

............ 99950 01

99960 01

99970 10

99980 01

99990 10

[10000rowsx16columns]df.drop(["Geography","Gender"],axis=1,inplace=True)

df.head()

CreditScoreAge TenureBalanceNumOfProductsHasCrCard\ 0 619 42 2 0.00 1 1

1 60841 183807.86 1 0

2 50242 8159660.80 3 1

3 69939 1 0.00 2 0

4 85043 2125510.82 1 1

IsActiveMemberEstimatedSalaryExitedFranceGermanySpainFemale\

0 1 101348.88 1 1 00

1

1 1 112542.58 0 0 01

1

2 0 113931.57 1 1 00

1

3 0 93826.63 0 1 0 0

1

4 1 79084.10 0 0 0 1

1

Male

00

10

20

30

40

x=df.drop('Exited',axis=1)

x

CreditScoreAge TenureBalanceNumOfProductsHasCrCard\ 0 619 42 2 0.00 1 1

1 60841 183807.86 1 0

2 50242 8159660.80 3 1

3 69939 1 0.00 2 0

4 85043 2125510.82 1 1

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

9995 77139 5 0.00 2 1

9996 516351057369.61 1 1

9997 70936 7 0.00 1 0

9998 77242 375075.31 2 1

9999 79228 4130142.79 1 1

IsActiveMemberEstimatedSalaryFrance GermanySpainFemaleMale

0 1 101348.88 1 0 0 1

0

1 1 112542.58 0 0 1 1

0

2 0 113931.57 1 0 0 1

0

3 0 93826.63 1 00 1

0

4 1 79084.10 0 01 1

0

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

...

9995 0 96270.64 1 00 0

1

9996 1 101699.77 1 00 0

1

9997 1 42085.58 1 00 1

0

9998 0 92888.52 0 10 0

1

9999 0 38190.78 1 00 1

0

[10000rowsx13columns]y=df['Exited']

y

0 1

1 0

2 1

3 0

4 0

..

99950

99960

99971

99981

99990

Name:Exited,Length:10000,dtype:int64 df.shape

(10000,14)

x.shape(10000,13)

y.shape (10000,)

fromsklearn.model\_selectionimporttrain\_test\_split x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=0)

x\_train.shape (8000,13)

x\_test.shape (2000,13)

y\_test.shape (2000,)

fromsklearn.preprocessingimportStandardScaler

sc=StandardScaler()

x\_train=sc.fit\_transform(x\_train)

x\_train

array([[0.16958176,-0.46460796,0.00666099,...,1.74309049,

1.09168714,-1.09168714],

[-2.30455945,0.30102557,-1.37744033,...,-0.57369368,

-0.91601335,0.91601335],

[-1.19119591,-0.94312892,-1.031415,...,-0.57369368,

1.09168714,-1.09168714],

...,

[0.9015152,-0.36890377,0.00666099,...,-0.57369368,

-0.91601335,0.91601335],

[-0.62420521,-0.08179119,1.39076231,...,1.74309049,

1.09168714,-1.09168714],

[-0.28401079,0.87525072,-1.37744033,...,-0.57369368,

1.09168714,-1.09168714]])

x\_test=sc.transform(x\_test)

x\_test

array([[-0.55204276,-0.36890377,1.04473698,...,-0.57369368,

1.09168714,-1.09168714],

[-1.31490297,0.10961719,-1.031415,...,-0.57369368,

1.09168714,-1.09168714],

[0.57162971,0.30102557,1.04473698,...,1.74309049,

1.09168714,-1.09168714],

...,

[-0.74791227,-0.27319958,-1.37744033,...,1.74309049,

-0.91601335,0.91601335],

[-0.00566991,-0.46460796,-0.33936434,...,-0.57369368,

-0.91601335,0.91601335],

[-0.79945688,-0.84742473,1.04473698,...,-0.57369368,

-0.91601335,0.91601335]])