[81]: [81]:	data=pd.read_csv("E:\\Data Analyti data Loan_ID Gender Married Dependence 0 LP001002 Male No 1 LP001003 Male Yes			plicantIncome Coa 5849 4583	0.0	Dunt Loan_Amount_Term NaN 360.0 28.0 360.0	1.0	Urban	Loan_Status Y N	
	2 LP001005 Male Yes 3 LP001006 Male Yes 4 LP001008 Male No 609 LP002978 Female No	0 Graduate 0 Not Graduate 0 Graduate 0 Graduate	Yes No No No	3000 2583 6000 2900	2358.0 1 0.0 1 	360.0 20.0 360.0 41.0 360.0 71.0	1.0 1.0 	Urban Urban Urban Rural	Y Y Y	
	610 LP002979 Male Yes 611 LP002983 Male Yes 612 LP002984 Male Yes 613 LP002990 Female No 614 rows × 13 columns	3+ Graduate 1 Graduate 2 Graduate 0 Graduate	No No No Yes	4106 8072 7583 4583	240.0 2 0.0 1	40.0 180.0 53.0 360.0 37.0 360.0 33.0 360.0	1.0	Rural Urban Urban Semiurban	Y Y Y N	
[54]: [54]:	data.head() Loan_ID Gender Married Depender 0 LP001002 Male No 1 LP001003 Male Yes	nts Education Self 0 Graduate 1 Graduate	_Employed Applic No No	cantIncome Coapp 5849 4583	0.0 Na		Credit_History Pr 1.0 1.0	roperty_Area Lo Urban Rural	pan_Status Y N	
[56] :	2 LP001005 Male Yes 3 LP001006 Male Yes 4 LP001008 Male No	0 Graduate0 Not Graduate0 Graduate	Yes No No	3000 2583 6000	0.0 66 2358.0 120 0.0 141	0 360.0	1.0 1.0 1.0	Urban Urban Urban	Y Y Y	
[56] :		0 592.000000 8 146.412162 9 85.587325		64.000000 0.842199 0.364878 0.000000						
	25% 2877.500000 0.000000 50% 3812.500000 1188.500000 75% 5795.000000 2297.250000		360.00000 360.00000 360.00000 480.00000	1.000000 1.000000 1.000000 1.000000						
	data.info() <class #="" 'pandas.core.frame.dataframe="" (total="" 0="" 13="" 613="" 614="" co<="" column="" columns="" columns):="" data="" entries,="" non-null="" rangeindex:="" td="" to=""><td>ount Dtype</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></class>	ount Dtype								
	1 Gender 601 non-nul 2 Married 611 non-nul 3 Dependents 599 non-nul 4 Education 614 non-nul 5 Self_Employed 582 non-nul 6 ApplicantIncome 614 non-nul 7 CoapplicantIncome 614 non-nul 8 LoanAmount 592 non-nul 9 Loan_Amount_Term 600 non-nul	object object object object object object int64 float64								
[58]:	10 Credit_History 564 non-nul 11 Property_Area 614 non-nul 12 Loan_Status 614 non-nul dtypes: float64(4), int64(1), objec memory usage: 62.5+ KB	1 float64 1 object 1 object								
	Gender 13 Married 3 Dependents 15 Education 0 Self_Employed 32 ApplicantIncome 0 CoapplicantIncome 0 LoanAmount 22 Loan_Amount_Term 14									
[82]:	Credit_History 50 Property_Area 0 Loan_Status 0 dtype: int64 data["LoanAmount"]=data["LoanAmound data["Credit_History"]=data["Credit data				an())					
[82]:	Loan_ID Gender Married Dependence 0 LP001002 Male No 1 LP001003 Male Yes 2 LP001005 Male Yes 3 LP001006 Male Yes	0 Graduate 1 Graduate 0 Graduate 0 Not Graduate	No No Yes No	5849 4583 3000 2583	0.0 146.412 1508.0 128.000 0.0 66.000 2358.0 120.000	000 360.0 000 360.0	1.0 1.0 1.0	Urban Rural Urban	Y N Y Y	
	4 LP001008 Male No 609 LP002978 Female No 610 LP002979 Male Yes	0 Graduate 0 Graduate 3+ Graduate	No No No	6000 2900 4106	0.0 141.000 0.0 71.000 0.0 40.000	000 360.0 000 360.0	1.0 1.0	Urban Rural	Y Y	
	611 LP002983 Male Yes 612 LP002984 Male Yes 613 LP002990 Female No 614 rows × 13 columns	1 Graduate2 Graduate0 Graduate	No No Yes	7583 4583	240.0 253.000 0.0 187.000 0.0 133.000	000 360.0	1.0	Urban Urban Semiurban	Y Y N	
[60]: [60]:	data.isnull().sum() Loan_ID 0 Gender 13 Married 3 Dependents 15 Education 0 Self_Employed 32 ApplicantIncome 0									
[61] :	CoapplicantIncome 0 LoanAmount 0 Loan_Amount_Term 14 Credit_History 0 Property_Area 0 Loan_Status 0 dtype: int64 data.shape									
[83]: [17]:	<pre>(614, 13) data.dropna(inplace=True) data.shape (542, 13)</pre>									
[76]: [76]:	data.isnull().sum() Loan_ID									
[84]:	CoapplicantIncome 0 LoanAmount 0 Loan_Amount_Term 0 Credit_History 0 Property_Area 0 Loan_Status 0 dtype: int64 #DATA VISUALIZATION									
	<pre>plt.figure(figsize=(100,50)) sns.set(font_scale=5) plt.subplot(331) sns.countplot(x="Gender", hue=data[plt.subplot(332) sns.countplot(x="Married", hue=data] plt.subplot(333)</pre>	a["Loan_Status"],da	ata=data)							
[84]:	<pre>sns.countplot(x="Education", hue=da plt.subplot(334) sns.countplot(x="Self_Employed", hu plt.subplot(335) sns.countplot(x="Property_Area", he </pre> <pre><axes: ,="" pre="" xlabel="Property_Area" ylabel<=""></axes:></pre>	ne=data["Loan_Statu nue=data["Loan_Stat	us"],data=data))						
	300 250 200 150 100	Loa	an_Status Y N	250 200 tuno 150			Loan_Status Y N	300 250 200 100 100		Loan_S
	100 50 0 Male Gen		an_Status	50 0	No	Married Yes		100 50 0		ate Not Graduate Education
1	250 250 200 3 150 100 50	1	Y	125 100 75 50 25		Y N				
[27]: [27]:	O No Self_Em	nts Education Self		cantIncome Coapp	olicantIncome LoanAmou	operty_Area			_ _	
	0 LP001002 Male No 1 LP001003 Male Yes 2 LP001005 Male Yes 3 LP001006 Male Yes 4 LP001008 Male No	0 Graduate1 Graduate0 Graduate0 Not Graduate0 Graduate	No No Yes No	5849 4583 3000 2583 6000	0.0 146.41216 1508.0 128.00000 0.0 66.00000 2358.0 120.00000 0.0 141.00000	0 360.0 0 360.0 0 360.0	1.0 1.0 1.0 1.0	Urban Rural Urban Urban Urban Urban	Y N Y Y	
	<pre>#Replacing variable values to numb data["Loan_Status"].replace("Y",1, data["Loan_Status"].replace("N",0, C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas</pre>	inplace=True) inplace=True) ernel_7536\1637887 3.0. This inplace	054.py:2: Futu method will ne	reWarning: A va	alue is trying to be se the intermediate	set on a copy of a I object on which we as	DataFrame or a	Series throug lues always b	th chained assignment as a copy.	
	<pre>object. data["Loan_Status"].replace("Y",1 C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas For example, when doing 'df[col].me</pre>	,inplace=True) ternel_7536\1637887 3.0. This inplace	054.py:3: Futu method will ne	reWarning: A va ver work becaus	alue is trying to be se the intermediate	set on a copy of a I object on which we an	DataFrame or a	Series throug lues always b	nh chained assignmehaves as a copy.	
	object. data["Loan_Status"].replace("N",0	,inplace=True) ernel_7536\1637887 se)`. To opt-in to ,inplace=True)	054.py:3: Futu	reWarning: Dowr	ncasting behavior in	`replace` is depreca	ated and will			on. To retain the old behavior, expl
[93]:	Loan_Status 1 422 0 192 Name: count, dtype: int64 data.Married = data.Married.map({" data["Married"].value_counts()	'Yes":1, "No":0})								
	Married 1 355 0 187 Name: count, dtype: int64 data Loan_ID Gender Married Dependence	dents Education S	Self_Employed Ap	plicantIncome Coa	applicantIncome LoanAm	ount Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	
	0 LP001002 Male No 1 LP001003 Male Yes 2 LP001005 Male Yes 3 LP001006 Male Yes	0 Graduate1 Graduate0 Graduate0 Not Graduate	No No Yes No	5849 4583 3000 2583	1508.0 1 0.0	NaN 360.0 28.0 360.0 66.0 360.0 20.0 360.0	1.0	Urban Rural Urban Urban	Y N Y	
	4 LP001008 Male No 609 LP002978 Female No 610 LP002979 Male Yes 611 LP002983 Male Yes	0 Graduate 0 Graduate 3+ Graduate 1 Graduate	No No No	6000 2900 4106 8072	0.0 0.0	41.0 360.0 71.0 360.0 40.0 180.0 53.0 360.0	 1.0 1.0	 Rural Rural	Y Y Y	
	612 LP002984 Male Yes 613 LP002990 Female No 614 rows × 13 columns	2 Graduate 0 Graduate	No Yes	7583 4583	0.0 1	37.0 360.0 33.0 360.0	1.0		Y N	
86]: [86]:	Loan_ID Gender Married Dependence O LP001002 Male No 1 LP001003 Male Yes 2 LP001005 Male Yes	dents Education S 0 Graduate 1 Graduate 0 Graduate	Self_Employed App No No Yes	5849 4583 3000	0.0 146.412 1508.0 128.000 0.0 66.000	000 360.0	1.0	Urban	Loan_Status 1 0 1	
	3 LP001006 Male Yes 4 LP001008 Male No 609 LP002978 Female No	0 Not Graduate 0 Graduate 0 Graduate	No No No	2583 6000 2900	2358.0 120.000 0.0 141.000 0.0 71.000	000 360.0 000 360.0	1.0 1.0	Urban Rural	1 1 1	
	 610 LP002979 Male Yes 611 LP002983 Male Yes 612 LP002984 Male Yes 613 LP002990 Female No 542 rows × 13 columns 	3+ Graduate 1 Graduate 2 Graduate 0 Graduate	No No No Yes	4106 8072 7583 4583	0.0 40.000 240.0 253.000 0.0 187.000 0.0 133.000	000 360.0 000 360.0	1.0	Rural Urban Urban Semiurban	1 1 0	
	<pre>data["Gender"].replace("Male",1,in data["Gender"].replace("Female",0, data C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas</pre> For example, when doing 'df[col].me	inplace=True) dernel_7536\1891471 3.0. This inplace	method will ne	ver work becaus	se the intermediate	object on which we as	re setting va	lues always b	ehaves as a copy.	
	data["Gender"].replace("Male",1,iC:\Users\hp\AppData\Local\Temp\ipykThe behavior will change in pandas For example, when doing 'df[col].me	ernel_7536\1891471 3.0. This inplace	method will ne	ver work becaus	se the intermediate	object on which we as	re setting va	lues always b	ehaves as a copy.	
	<pre>call `result.infer_objects(copy=Fal data["Gender"].replace("Female",0</pre>	ernel_7536\1891471 se)`. To opt-in to ,inplace=True)	the future be	havior, set `po	d.set_option('future	no_silent_downcastir	ng', True)`			on. To retain the old behavior, expli
	1 LP001003 1 Yes 2 LP001005 1 Yes 3 LP001006 1 Yes 4 LP001008 1 No	 Graduate Graduate Not Graduate Graduate 	No Yes No No	4583 3000 2583 6000	1508.0 128.000 0.0 66.000 2358.0 120.000 0.0 141.000	000 360.0 000 360.0	1.0 1.0 1.0 1.0	Rural Urban Urban Urban	0 1 1	
	609 LP002978 0 No 610 LP002979 1 Yes 611 LP002983 1 Yes 612 LP002984 1 Yes	0 Graduate 3+ Graduate 1 Graduate 2 Graduate	No No No No	2900 4106 8072 7583	0.0 71.000 0.0 40.000 240.0 253.000 0.0 187.000	000 180.0 000 360.0	1.0	Rural Rural Urban Urban	1 1 1	
[88]:	613 LP002990 0 No 542 rows × 13 columns data["Dependents"].replace("3+",3,data C:\Users\hp\AppData\Local\Temp\ipyk		Yes	4583	0.0 133.000		0.0	Semiurban	o o o o o o o o o o o o o o o o o o o	ment using an inplace method
	The behavior will change in pandas	3.0. This inplace the thod(value, inplace the thod)	method will ne	ver work becaus	se the intermediate od({col: value}, inp	object on which we as	re setting va	lues always b	ehaves as a copy.	
	 0 LP001002 1 No 1 LP001003 1 Yes 2 LP001005 1 Yes 3 LP001006 1 Yes 	0 Graduate1 Graduate0 Graduate0 Not Graduate	No No Yes No	5849 4583 3000 2583	0.0 146.412 1508.0 128.000 0.0 66.000 2358.0 120.000	360.0 000 360.0 000 360.0	1.0		1 0 1	
	4 LP001008 1 No	0 Graduate 0 Graduate 3 Graduate 1 Graduate	No No No No	6000 2900 4106 8072	0.0 141.000 0.0 71.000 0.0 40.000 240.0 253.000	 000 360.0 000 180.0	1.0	Urban Rural Rural Urban	1 1 1	
[89]:	612 LP002984 1 Yes 613 LP002990 0 No 542 rows × 13 columns data["Education"].replace("Graduat	2 Graduate 0 Graduate	No Yes	7583 4583	0.0 187.000 0.0 133.000			Urban Semiurban	0	
	<pre>data["Education"].replace("Not Gra data["Education"].value_counts() C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas</pre>	ernel_7536\2633681	Frue) .571.py:1: Futu method will ne	ver work becaus	se the intermediate	object on which we as	re setting va	lues always b	ehaves as a copy.	
	object.	ernel_7536\2633681 3.0. This inplace in thod(value, inplace)	571.py:2: Futu method will ne e=True)', try	ver work becaus	se the intermediate	object on which we as	re setting va	lues always b	ehaves as a copy.	
	data["Education"].replace("Not Gr C:\Users\hp\AppData\Local\Temp\ipyk call `result.infer_objects(copy=Fal data["Education"].replace("Not Gr Education 1 425 0 117 Name: count, dtype: int64	ernel_7536\2633681 se)`. To opt-in to	571.py:2: Futu the future be					be removed i	n a future versio	on. To retain the old behavior, expli
		ernel_7536\2004937 3.0. This inplace	method will ne	ver work becaus	se the intermediate	object on which we as	re setting va	lues always b	ehaves as a copy.	
	data["Self_Employed"].replace("YeC:\Users\hp\AppData\Local\Temp\ipykThe behavior will change in pandas For example, when doing 'df[col].meobject.	ernel_7536\2004937 3.0. This inplace	048.py:2: Futu method will ne	ver work becaus	se the intermediate	object on which we an	re setting va	lues always b	ehaves as a copy.	
	data["Self_Employed"].replace("No C:\Users\hp\AppData\Local\Temp\ipyk call `result.infer_objects(copy=Fal data["Self_Employed"].replace("No Self_Employed 0 467 1 75 Name: count, dtype: int64	ernel_7536\2004937 se)`. To opt-in to			_			be removed i	n a future versio	on. To retain the old behavior, expl
	data["Property_Area"].replace("Urb data["Property_Area"].replace("Rur data["Property_Area"].replace("Sem data["Property_Area"].value_counts C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas	ral",1,inplace=True niurban",2,inplace= s() ternel_7536\3076698 3.0. This inplace	928.py:1: Futu method will ne	ver work becaus	se the intermediate	object on which we an	re setting va	lues always b	ehaves as a copy.	
	<pre>object. data["Property_Area"].replace("Ur C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas For example, when doing 'df[col].me</pre>	ban",0,inplace=Trucernel_7536\3076698	ne) 1928.py:2: Futu method will ne	reWarning: A va ver work becaus	alue is trying to be se the intermediate	set on a copy of a I object on which we an	DataFrame or a	Series throug lues always b	ch chained assignmeter as a copy.	
	object. data["Property_Area"].replace("Ru C:\Users\hp\AppData\Local\Temp\ipyk The behavior will change in pandas	eral",1,inplace=True ernel_7536\3076698 3.0. This inplace	ne) 1928.py:3: Futu method will ne	reWarning: A va ver work becaus	alue is trying to be se the intermediate	set on a copy of a I object on which we an	DataFrame or a	Series throug lues always b	nh chained assignmehaves as a copy.	ment using an inplace method.
	data["Property_Area"].replace("Se C:\Users\hp\AppData\Local\Temp\ipyk call `result.infer_objects(copy=Fal data["Property_Area"].replace("Se Property_Area 2 209 0 174	ernel_7536\3076698 se)`. To opt-in to	928.py:3: Futu the future be		_			be removed i	n a future versio	on. To retain the old behavior, expl
[95]: [95]:		nts Education Self_Em	nployed Applicant 0	Income Coapplica 5849	untIncome LoanAmount 0.0 146.412162	Loan_Amount_Term Cred 360.0	dit_History Prope	erty_Area Loan_ 0	_Status 1	
	0 LP001002 1 0 1 LP001003 1 1 2 LP001005 1 1 3 LP001006 1 1 4 LP001008 1 0	0 1 1 1 0 1 0 0 0 1	0 0 1 0	5849 4583 3000 2583 6000	0.0 146.412162 1508.0 128.000000 0.0 66.000000 2358.0 120.000000 0.0 141.000000	360.0 360.0 360.0 360.0	1.0 1.0 1.0 1.0	0 1 0 0	0 1 1	
	#Splitting Data x=data.iloc[1:542,1:12].values x array([[1, 1, '1',, 360.0, 1.0 [1, 1, '0',, 360.0, 1.0 [1, 1, '0',, 360.0, 1.0,	0, 0], 0, 0],								
	<pre>y=data.iloc[1:542,12].values y array([0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,</pre>	0, 0], 0, 2]], shape=(541, 1, 1, 0, 1, 1, 1,	0, 0, 0, 1, 0,	1, 1,						
		1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1,						
	1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1,						
	0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0])	0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	0, 1, 1, 0, 1, 0, 0, 1, 0, 1,						
108	x_train, x_test, y_train, y_test=train print("Shape of x_train :", x_train print("Shape of x_test :", x_test.s print("Shape of y_train :", y_train print("Shape of y_test :", y_test.s Shape of x_train : (405, 11) Shape of x_test : (136, 11) Shape of y_train : (405,)	n.shape) shape) n.shape)	size=0.25)							
109	Shape of x_test : (136, 11) Shape of y_train : (405,) Shape of y_test : (136,) from sklearn.linear_model import I model=LogisticRegression() model LogisticRegression (2)	LogisticRegression								
112	▼ LogisticRegression		.ib\site-packag	es\sklearn\line	ear_model_logistic.	oy:473: ConvergenceWa	arning: lbfgs	failed to co	onverge after 100	<pre>iteration(s) (status=1):</pre>
	Increase the number of iterations to you might also want to scale the da https://scikit-learn.org/stable please also refer to the documentat https://scikit-learn.org/stable n_iter_i = _check_optimize_result LogisticRegression	o improve the converta as shown in: c/modules/preproces ion for alternative c/modules/linear_mo	ssing.html re solver optio	ns:						
	► Parameters y_pred=model.predict(x_test)	acy_score								
112	<pre>from sklearn.metrics import accura accuracy_score(y_pred,y_test)</pre>									
112 134 137 142)								
134 137 137 157 160	<pre>accuracy_score(y_pred,y_test) 0.8382352941176471 print("Predicted value:",y_pred[2] print("Test value:",y_test[2]) Predicted value: 0 Test value: 0</pre>	1, 1, 1, 1, 0,	1, 1, 1, 1, 0,	1, 1,						