]:	<pre>import numpy a import matplot import seaborn import missing from scipy imp import sklearn %matplotlib in df = pd.read_c df.shape</pre>	gno as msr port stats n nline		index_c	col = 0)					
]:	(500, 8) df.head() Email 0 adkv@ota.com	89280 Ma	rk Lane\nNew Jo		Credit Card 3544288738428794	Avg. Session Length	Time on App	Time on Website	Length Members 4.582	ship Sp
	 gjun@syj.com qjyr@pkk.com jkiu@xsb.com stvb@niy.com 	62008 Ada 950 Tami I	sland\nLake Ain M Squares\nNorth	Pamela, 1Y 30677 neeview, 1T 93614	6546228325389133 4406395951712628314 30334036663133 3582080469154498	32.926272 34.000915 35.305557 34.330673	12.330278 14.717514	38.268959 38.110597 37.721283 38.536653	3.164 4.604 3.620 4.946	543 488.547! 179 582.852:
	df.tail() Email 495 xskz@gwj.com 496 awrc@iok.com	7083 Wall	ace Rest\nNew [*] Christopher Gar Carrieberg	70240	3020674202308	Length 35 34.237660	1 App		Lengt Member 4.246 4.076	ship Sp 6573 574.8474
	<pre>497 pndt@jyr.com 498 zvtz@onj.com 499 phqb@nlg.com df.describe()</pre>	1 424 Mark 1	5568 Robert S 030\nTurners	NH 46418 Station Apt. Stad, GA 9 Prelichester, TX 09088	3621809248806	34.322501 6 34.715981	1 13.418808	3 37.840086 3 36.771016	5.458 2.836 3.235	6485 457.469!
	count 5.000000e+0 mean 3.706324e+1 std 1.235588e+1 min 5.018057e+1 25% 3.683275e+1 50% 3.513612e+1 75% 4.777131e+1	D2 17 18 11 13	500.000000 34.053194 0.992563 30.532429 33.341822 34.082008 34.711985	500.00000 13.05248 0.9942 9.50819 12.38819 12.98323 13.75388	500.00000 500.000000 88 38.060445 16 1.010489 52 34.913847 53 37.349257 31 38.069367	500	0.000000 4.033462 0.999278 0.769901 3.430450 4.033975 4.626502	500.000 500.314 79.314 257.670 446.038 499.887 550.313	0000 4038 4782 0582 8277 7875	
	Session Length, Timvalue. msno.bar(df, f	to be credit ne on app, -	Time on Webs	site, Lengt	nformation about dis th of Membership wi	tributions carri Il be worth loo	king at as p	_	mporal v	_
	1.0 0.8 0.6 0.4	500	500	\$°	400 400	400	3	00 00 00		
	0.2 0.0 Email Addi	credit.C	and Session Length	i fine on	ADP Time on Website	Veally Amount Spe		00		
	This implies no miss plt.figure(figure sns.scatterplc plt.show()	gsize = (1	10,10)) df, x = "T		Website", y = "Yes		Spent",	alpha = 0.7	7).set(title = "Ti
	Farly Amount Spent 000						••			
	400 - 300 -						•			
]:	plt.figure(fig sns.scatterplo plt.show()		df, x = "T	Time on	38 39 Website App", y = "Yearly yearly amount spent		41	na = 0.7).se	et(titl	e = "Time o
	- 000 -									
	Mearly Amount Spent						•			
]:	plt.figure(figsns.scatterploplt.show()	gsize = (1	10,10))	Time o	13 14 on App f Membership", y	15 = "Time on	16 App", al	pha = 0.7).s	set(tit	:le = "Lengt
	16 - 15 -		Length of me	embership	versus time on app					
	Time on App						•			
1.	10 -	2	3 L	4 ength of Me	5 embership	6	7			
	sns.pairplot(d		rid at 0x24	00aa008	e0>	- CONTROL OF THE PARTY OF THE P	(00000000000000000000000000000000000000	(0.03 a) ((0.10.)0 (0.00)		
	0 1 37 0 436 0 18 0 18 0 19 0 10 0								6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6	
	11 10 10 10 10 10 10 10 10 10 10 10 10 1									
	Rearly Amount Spent Length of Membership 2009 1 1 2 1 6 6 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9									
0 0	Credit Ca Length of Members corr = df.corr plt.figure(fig	ship is most (() gsize = (1 prr, annot	10,10)) t = True).se	ith Yearly	10 12 14 16 Time on App amount spent. e = "Correlation	36 38 Time on Web	40 osite	2 4 Length of Member	6 sship	400 600 Yearly Amount Sp
	Credit Card Credit Card Avg. Session Length		-0.024	-0.033 -0.028		0.016 -0.03		- 1.0 - 0.8		
	Time on App Time on Website		-0.028 -0.035	0.082		.029 0.5 .048 -0.00		- 0.6 - 0.4		
	Length of Membership Yearly Amount Spent		Avg. Session Length - 00 09 09 09 09 09 09 09 09 09 09 09 09	0.029 - ddb - 0.5		ingth of Membership - 1880		- 0.2 - 0.0		
	Average Session ler is the most related spent. Physical add if this persists so it	it Card and ngth and Ti to yearly ar ress could i also should	Time on Web me on App ha mmount spen f it was region be dropped.	site shoul ave slight t. Non Nu n based, b	Id be removed from positive correlation lumerical Columns shout that is beyond the	the dataset as out appear to bould also be re e scope of this	there is no be noise, be moved as e	ut it is clear the email cannot p	at Lengt oredict y	th of membersl early amount
]:	plt.figure(fig	gsize = (1 ot(data =	10,10)) df, x = "L	ength oi	f Membership", y	= "Yearly A	mount Spe	ent", alpha	= 0.7)	.set(title
	Fearly Amount Spent									
	400 -									
	Based on the EDA a	and null ana	ction a	nd F	Pre-proces ssary columns for the	regression	7			
	should be 101 \ As sklearn's StandardS	test_split to we will be of Scaler for sc stion: https	o split the data doing gradien alling the X of ://stackoverflo	a set into it descent f training ow.com/q	training and test sets as well as some oth and test sets. But do questions/38780302/	er regression to n't do it for y(t	echnique, s arget) train	caling the dat and test. [For	a set is i help, yo	mportant. So, ou can see the
	training-data from sklearn.m	nodel sele	:4].values :].values r() transform(X) = train_	_test_split(X_sca	iled, y, tes	t_size=0	.3, random_s		
]:	from sklearn.m from sklearn.m X = df_clean.i Y = df_clean.i Scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m	preprocess loc[:, 0: loc[:, 4: dardScaler aler.fit_t st, y_train	Sklearn''s Lin	_	ssion (example in the				ebcours	
]:	from sklearn.m from sklearn.m X = df_clean.i Y = df_clean.i Scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m	g Line model using show the contest data atter plot the metrics to pure linear_model usiness linear_model usiness	Sklearn''s Line pefficients and at shows the N rint the value ne coefficient a del import mean_	d intercep Y test on a of MAE, N and which	ssion (example in the ot x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs	linear regress d in y-axis (see documen	ion slide/co	olab links in wo	s)	es)
]:	from sklearn.m from sklearn.m X = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegr reg.fit(X_trai) LinearRegressic print("Best va Best value of t [500.53166775] [[25.76252659 3]	g Line model using show the contest data atter plot the netrics to pure linear_model atteries to pure linear_model atteries to pure linear_model atteries in the business discontinuous in the business discontinuous in the contest in the state of the contest in the state of the contest in the state of the contest in the	Sklearn's Line pefficients and at shows the N rint the value ne coefficient del import mport mean_ in) neta:",reg. 02 0.19220 _test)	d intercep I test on a of MAE, N and which LinearResquared intercep	egression error pt_, reg.coef_, seg	d in y-axis (see documen to which featu	ion slide/co	olab links in wo	s)	es)
]:	from sklearn.m from sklearn.m X = df_clean.i Y = df_clean.i Y = df_clean.i Scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.n reg=LinearRegr reg.fit(X_trai LinearRegressic print("Best va Best value of t [500.53166775] [[25.76252659 3] y_pred = reg.p plt.figure(fig comparison = p y_test = y_test y_pred = y_pre comparison['tect comparison['tect comparison['predict c	g Line model using show the contest data atter plot the netrics to pure linear_model usiness inear_model usiness	Sklearn"s Line pefficients and at shows the national shows the nationa	test on a of MAE, Nand which LinearResquared interces 992 61.	ssion (example in the ot x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression error	d in y-axis (see documen to which featu	tation of skure and bas	clab links in we	s) plain any	y strategy that
	from sklearn.m from sklearn.m X = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegment (X_train) LinearRegment (X_train) print ("Best value of topic (S)	g Line model using show the contest data atter plot the netrics to pure linear_model usiness inear_model usiness	Sklearn"s Line pefficients and at shows the national shows the nationa	test on a of MAE, Nand which LinearResquared interces 992 61.	ssion (example in the ot x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression error pt_, reg.coef_, segundation of the other	d in y-axis (see documen to which featu	tation of skure and bas	clab links in we	s) plain any	y strategy that
]:	from sklearn.m from sklearn.m X = df_clean.i Y = df_clean.i Scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegr reg.fit(X_trai) LinearRegressic print("Best va Best value of t [500.53166775] [[25.76252659 3] y_pred = reg.p plt.figure(figure) comparison = p y_test = y_test y_pred = y_pred comparison['tect com	g Line model using show the contest data atter plot the netrics to pure linear_model usiness inear_model usiness	Sklearn"s Line pefficients and at shows the national shows the nationa	test on a of MAE, Nand which LinearResquared interces 992 61.	ssion (example in the ot x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression error pt_, reg.coef_, segundation of the other	d in y-axis (see documen to which featu	tation of skure and bas	clab links in we	s) plain any	y strategy that
	from sklearn.m from sklearn.m x = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegr reg.fit(X_train) LinearRegression print("Best value of t [500.53166775] [[25.76252659 3] y_pred = reg.p plt.figure(figure (figure of test) y_test = y_test y_pred = y_pred comparison['test) comparison['test) sins.scatterplo plt.show() 700 700 700 700 700	preprocess loc[:, 0: loc[:, 4: loc[:, 4: dardScaler aler.fit_t st, y_train Condel using show the contest data atter plot the netrics to pu Interpret the business linear_mod netrics in lession() lin, y_train lon() alue of the theta: 38.3285520 bredict(X_ location in	Sklearn's Line pefficients and at shows the and at shows the arithmeticient and at shows the arithmetic and at shows the arithmetic and at shows the arithmeti	test on a of MAE, Nand which which the squared squared squared absolute ore y_pred)	ssion (example in the out x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression _error pt_,reg.coef_,ser 17355707]] test", y = "predicted values edicted Values	d in y-axis (see documen to which featu	tation of skure and bas	clab links in we	s) plain any	y strategy that
	from sklearn.m from sklearn.m X = df_clean.i y = df	Coreprocess Cloc[:, 0: Cloc[:, 4: Cloc[:, 4: ClardScaler ClardScal	Sklearn's Line perficients and at shows the and at shows	test on a of MAE, Nand which which the squared squared squared squared absolute ore y_pred) y_pred) y_pred) y_pred) y_pred, \nR_squared squared square	ssion (example in the out x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression _error pt_,reg.coef_,ser 17355707]] test", y = "predicted values edicted Values	d in y-axis (see documento which feature) cted", alpha ich correspontation membership	tation of skure and base d to Avg. Shas the big	ession Length gest effect on	s) plain any	es) y strategy that n App, Time of
	from sklearn.m from sklearn.m from sklearn.m x = df_clean.i y = df_clean.i y = df_clean.i scaler = Stand x_scaled = sca x_train, x_tes 4 Training 1. Train a linear m 2. After training, s 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegr reg.fit(X_trainer) from sklearn.m reg=LinearRegr reg.fit(X_trainer) print("Best value of to	greprocess loc[:, 0: loc[:, 4: loc[:	Sklearn"s Lin Defficients and at shows the and at shows the and at shows the and rint the value he coefficient and del import del import moort mean in) heta:", reg. 10,10) heta:", reg. 20,19220 test) 10,10) heme() Series (y_test, regres, regres, rest, rest, regres, regress, regress	test on a of MAE, Nand which which the squared absolute ore y_pred) y_pred) y_pred) y_pred) y_pred) y_pred) y_pred) y_pred) y_pred) the squared absolute ore y_pred) y_pred) y_pred) y_pred) the squared absolute ore y_pred) the y_pred) the squared absolute ore y_pred) the y_pred)	ssion (example in the out x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs egression error pt_,reg.coef_,seg 17355707]] test", y = "predicted values edicted Values edicted Values 992 61.17355707], who is predicted, Length out greater (}".format	inear regress d in y-axis (see documen to which featu a '\n') cted", alph cted", alph different data to see the co ocess. Also, the training set	tation of skure and base as to Avg. Shas the big ength of many structures lab link I have a structures lab link I have a structures lab link I have a structure and lab link I have a structure at lab link I have a stru	eleasin's metricised on that expenses effect on the second embership.	Time of the modern the modern the modern the modern the modern the contract the modern t	es) y strategy that ex versus Pre night need to class and try to
	from sklearn.m from sklearn.m X = df_clean.i y = df_clean.i y = df_clean.i y = df_clean.i x_scaler = scat X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, 3 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m reg=LinearRegr reg.fit(X_trai) LinearRegressic print("Best value of t [500.53166775] [[25.76252659 3] y_pred = reg.p plt.figure(fig comparison = p y_test = y_tes y_pred = y_pre comparison['te comparis	preprocess loc[:, 0: loc[:, 4: dardScale dardS	Sklearn's Line perficients and at shows the value at shows the value are coefficient and at shows the value are coefficient and at shows the value are the state of the other state at shows the value are they very coefficient and their data and their data are they very coefficient and their data are they	test on a of MAE, Nand which which squared absolute ore y_pred) y_pred) y_pred, x = "to yersus Prediction of MAE, Nand which where y_pred) y_pred, x = "to yersus Prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand which was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of MAE, Nand was a to get an est_theta close to the yersus prediction of th	ssion (example in the state of	inear regress d in y-axis (see document to which feature to which feature to which feature to which feature to which less than Late of the see the cooperation of the see the se	tation of sk tream	ession Length gest effect on that expending the second sec	Time of the modern the slide	es) y strategy that ex versus Pre night need to class and try to
	from sklearn.m from sklearn.m from sklearn.m X = df_clean.i y = deren x_scaled = sca x_train, x_tes 4 Training 1. Train a linear m 2. After training, 3. Predict for the 4. Generate a sca 5. Use sklearn.m from sklea	greening and services in the state of the st	Sklearn"s Line Defficients and State shows the Norm Interest the value The coefficient of	test on a cof MAE, Nand which which squared absolute y_pred) y	ssion (example in the other than than than the other than than than than than than than than	inition regress d in y-axis (see documento which feature acted", alpha cted", alpha cted", alpha cteds. Also, the training set gression? d in y-axis (see documento	ion slide/contation of skure and base and a structures lab link I have code and a structure lab link I have code a structure lab link I have code a structure lab link I have code and a structure lab link I have code a structure lab li	ession Length gest effect on that experience is earn's metric seed on that experience is earn's metric seed on the estimate of	Test The of the model in the slide is side in the slide is side in the slide is side in the slide in the sli	es) y strategy that ex versus Pre night need to class and try to
	from sklearn.m from sklearn.m X = df_clean.i Y = df	general services in the state of the state o	Sklearn's Line Deficients and State shows the	dintercep dintercep dintercep diest on a of MAE, N and which LinearRe squared squared intercep 92 61. versus Pro nR_squa st) versus Pro versus Pro nR_squa st to get an est_theta close to the fits on a of MAE, N and equat fits on a of MAE, N and equat not of MAE, N and equat not of MAE, N nal equat	ssion (example in the of the o	inch correspon to which featu a different data to which less than L different data to see the co becas. Also, the training set gression? d in y-axis (see document XB_train_T. XB_train_T.	ion slide/contact the slide and base and slide	ession Length gest effect on lembership. such as to_nu ave shown you discussion in to establish axis = 1)	Test "Test	n App, Time or del while Time might need to class and try to will help. \
	from sklearn.m from sklearn.m from sklearn.m X = df_clean.i y = df_clean.i y = df_clean.i y = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, 3. Predict for the 4. Generate a sca 5. Use sklearn's 6. Interpretation: should help the from sklearn.m from sklearn.m freg_linearRegr reg_fit(X_trai) print("Best value of t [500.53166775] [125.76252659 3 y_pred = reg_f y_test = y_tes y_pred = y_pre comparison = p y_test = y_tes y_pred = y_pre comparison['tre comparison ['tre co	detrics in the state of the sta	Sklearn's Line Sklearn's Line Perficients and at shows the Norm at shows the Norm the value ne coefficient del import mport mean_ in) neta:", reg. 102 0.19220 Lest) 10,10) ame () Series (y_te = pad. Series comparison Test cor (y_test,	dintercep Intercep Itest on a Intercep Itest on a Intercep Itest on a Intercep Itest on a Ite	ssion (example in the of the state of the st	ininear regress d in y-axis (see documento which feature be a different data to which feature cted", alph training set gression? d in y-axis (see documento cted", alph training set gression? d in y-axis (see documento training set gression? d in y-axis (see documento training set gression? d in y-axis (see documento training set gression?	ion slide/contation of skure and base and structures lab link I have code and station of skure at the skure a	ession Length glearn's metric acted on that exp such as to_nu ave shown you discussion in the expectation of the second of the	Test Test The of the mode The slide The	n App, Time or del while Time night need to class and try to will help. \
	from sklearn.m from sklearn.m from sklearn.m from sklearn.m X = df_clean.i y = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear m 2. After training, 3. Predict for the 4. Generate a sca 5. Use sklearn's m 6. Interpretation: should help the from sklearn.m from sklearn.m from sklearn.m reg_LinearRegressic print ("Best value of t [500.53166775] [125.76252659 3 y_pred = reg.p plt.figure (fig comparison ['te	Control of the trick of the tri	Sklearn's Lin Defficients and at shows the norm at shows the norm at shows the norm at shows the norm and the import at shows the no	tintercep y test on y of MAE, N and which and which squared squared intercep y pred y, x = "t y and y y pred y pred y, x = "t y and y y pred y p	ssion (example in the oft state of the o	inear regress d in y-axis (see documen to which featu a='\n') a='\n') and the see the concess. Also, the training set t	ion slide/contain slide ion slide ion slide ion of skiller and base at a code and in the tain in the t	ession Length glearn's metric fied on that exp such as to_nu ave shown you discussion in the field on the state of the st	Time of the modern solution of the modern solution in the slide solution is the slide solution of the modern solution of the slide solution is the slide solution of the slide slide solution of the slide slide slide solution of the slide slide	n App, Time of del while Time might need to class and try to will help. \ 11], train_t 7, test_thet
	from sklearn.m from sklearn.m from sklearn.m from sklearn.m scalest est and x scalest est and controlled the from sklearn.m from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m from sklearn.m from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m from sklearn.m from sklearn.m from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m regalinearRegr reg.fit(x_trai from sklearn.m from	Control of the trick of the tri	Sklearn's Lin Defficients and at shows the norm at shows the norm at shows the norm at shows the norm and the import at shows the no	tintercep y test on y of MAE, N and which and which squared squared intercep y pred y, x = "t y and y y pred y pred y, x = "t y and y y pred y p	ssion (example in the of the state of the st	inear regress d in y-axis (see documen to which featu a='\n') a='\n') and the see the concess. Also, the training set t	ion slide/contain slide ion slide ion slide ion of skiller and base at a code and in the tain in the t	ession Length glearn's metric fied on that exp such as to_nu ave shown you discussion in the field on the state of the st	Time of the modern solution of the modern solution in the slide solution is the slide solution of the modern solution of the slide solution is the slide solution of the slide slide solution of the slide slide slide solution of the slide slide	n App, Time of del while Time might need to class and try to will help. \ 11], train_t 7, test_thet
	from sklearn.m from sklearn.m from sklearn.m from sklearn.m x = df_clean.i y = df_clean.i y = df_clean.i y = df_clean.i scaler = Stand X_scaled = sca X_train, X_tes 4 Training 1. Train a linear 2. After training. 3. Predict for the 4. Generate as ca 5. Use sklearn's 6. Interpretation: should help the from sklearn.m from sklearn.m from sklearn.m reg.fit(X_trai LinearRegressic print ("Best va Best value of to [500,53166775] [125,76252659 g Y_pred = reg.fi plt.figure (fig comparison = fig y_test = y_tes y_pred = y_pred y_pred = y_pred y_pred = y_pred y_pred = y_pred plt.show() 5 Norma (while solving this, y reshape, perform Tr comparison['pr sns.scatterple plt.show() 5 Norma (while solving this, y reshape, perform Tr comparison['pr sns.scatterple plt.show() 5 Norma (while solving this, y reshape, perform Tr comparison['pr sns.scatterple plt.show() 700-	preprocess loc[; 0] loc[; 4] l	Sklearn's Lin Defficients and at shows the norm at shows the norm at shows the norm at shows the norm and the import at shows the no	tintercep y test on y of MAE, N and which and which squared squared intercep y pred y, x = "t y and y y pred y pred y, x = "t y and y y pred y p	ssion (example in the of the state of the st	inear regress d in y-axis (see documen to which featu a='\n') a='\n') and the see the concess. Also, the training set t	ion slide/contain slide ion slide ion slide ion of skiller and base at a code and in the tain in the t	ession Length glearn's metric fied on that exp such as to_nu ave shown you discussion in the field on the state of the st	Time of the modern solution of the modern solution in the slide solution is the slide solution of the modern solution of the slide solution is the slide solution of the slide slide solution of the slide slide slide solution of the slide slide	n App, Time of del while Time might need to class and try to will help. \ 11], train_t 7, test_thet
	training-data from sklearn.m from sklearn.m from sklearn.m from sklearn.m from sklearn.m from sklearn.m scaler esta x scaled e sca X scaled e sca X train, x tes 4 Training 1. Train a linearing 3. Predict for the 4. Generate a sca 5. Use sklearn.m regeniearnem re	guinerics in a control of the contro	Sklearn's Lin pefficients and at shows the \(\) at	dintercep dintercep fetston of MAE, N of MAE, N and which LinearRe squared squared intercep 992 61. (y_pred) y_pred) y_pred) y_pred fetston of MAE, N of MAE, N nal equat fetston of MAE, N nal equat 1, which in t_T_dot Theta 3: Theta 4: Theta 3: Theta 4: Theta 5: Theta	ssion (example in the state of	din y-axis (see documento which feature be different data ceted", alph ceted", alph din, X3_trai autoses than L din y-axis (see documento din	ion slide/contation of skiller and base and base and base are and structures lab link I have a code and tation of skiller and base are code and tation of skiller and	ession Length gest effect on lembership. such as to_nu embership. such as to_nu discussion in to clearn's metric	Time of the modern solution of the modern solution in the slide solution is the slide solution of the modern solution of the slide solution is the slide solution of the slide slide solution of the slide slide slide solution of the slide slide	n App, Time of del while Time might need to class and try to will help. \ 11], train_t 7, test_thet
	training-data from sklearn.r. from sklearn.r. from sklearn.r. x = df clean.i y =	metrics immetrics of interpret the business distriction interpret the business districts of interpret the districts of interpret the business districts of interpret the districts of interpret	Sklearn"s Lin perficients and at shows the and at shows t	dintercep dintercep for test on the service of MAE, Nand whick squared absolute squared absolute squared wersus Proversus Proversus Proversus Provent for MAE, Nand equat at the squared at the squa	ssion (example in the state of	din y-axis (see document to which feature) mae, mse, resolution along the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression? din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression) din y-axis (see document to see the concess. Also, the training set pression)	ion slide/contain of slide in a structures lab link I have a code and station of slide in the tain the tain in the	ession Length gest effect on embership. such as to_nu ave shown you discussion in to elearn's metric clearn's metric such as to_nu ave shown you discussion in to elearn's metric clearn's metric	and the modern sheet a [1] The the ta [1] The ta [1] The ta [1]	y strategy that In App, Time or App, Time or Apply the or In App, Time or Apply the or In App, Time or
	training-data from sklearn.# from sklearn.# from sklearn.# from sklearn.# x = dclean.i y = der elean.i from sklearn.# reg=inearRegresic inearRegresic inearRegresic print("Best value of to (500.53166755) [Izo.53166755] [Izo.53166755] [Izo.53166755] [Izo.53166755] [Izo.53166755] [Izo.53166765] y _ pred = reg. f plt.figure (fig comparison) of the comparison of th	general control of the control of th	Sklearn"s Lin Scheric Lin Sefficients and sefficients and sefficients and sefficients and sefficients are sefficients. Sefficients and sefficients are sefficients. Sefficients are sefficients. Sefficients are sefficients. Sefficients and find by the sefficient sefficients. Sefficients are sefficients. Sefficients are shows the variety of the sefficients. Sefficients are sefficients. Se	squared absolute ore y_pred) y_pred) y_pred, \nR_squared est_theta close to the squared est_t	ssion (example in the station (example in the the x-axis and y predicte MSE, RMSE, and R^2 h coefficient belongs agression gersor pt_, reg.coef_, seg 17355707]] best", y = "predicted values edicted Values st 600 at 600 at 600 x-axis and y predicted, Length o e some effect, but m ataframe into various ould recommend you in idea during this pre values based on the ne sklearn's linear reg ared: ()".format x-axis and y predicted x-axis and y predicted ye some effect, but m ataframe into various ould recommend you in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea during this pre values based on the ne sklearn's linear reg in idea duri	din y-axis (see documento which features which feat	ion slide/contain slide/contai	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	### Training ### T	actrics impered to the control of th	Sklearn's Lin befficients and at shows the variet the value are coefficient. The value are coefficient and at shows the variet the value are coefficient. The value are coefficient are comparison. Test: 400 400 Apport mean_more register. Apport mean_more regist	dintercep dintercep for the construction of MAE, N and which squared absolute appred) y_pred) y_pred) y_pred) y_pred, x = "t for MAE, N and est_theta close to the for MAE, N and es	ssion (example in the station of the state o	din y-axis (see documen to which feature	ion slide/contain slide/contai	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	training-data from sklearn.m from sklearn.m from sklearn.m x adf_clear.i x adf_clear.i x adf_clear.i x ale clear.i x ale clear.	control of the contro	Sklearn's Lin befficients and shows the variet the value at shows the variet the value are coefficient. In the value are co	squared absolute or y predly y	ssion (example in the state of	din y-axis (see documen to which feature	tation of sk tation of sk tation of sk tare and bas a = 0.7) d to Avg. S has the big ength of m structures lab link I has code and tation of sk tation of sk tation of sk station of sk a = 0.7) See sample you can ge colab link: p=sharing axis tation of sk station of sk	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	training-data from sklearn.m from sklearn.m from sklearn.m from sklearn.m x = df_clean.i y = df_clean.i y = df_clean.i x = scas	preprocess	Sklearn's Lin perficients and perficients and at shows the Normal to the value are coefficient. The coefficient are coefficient are coefficient. The coefficient are	styperdiger of the styperdiger o	ssion (example in the station of the state o	din y-axis (see documen to which feature	tation of sk tation of sk tation of sk tare and bas a = 0.7) d to Avg. S has the big ength of m structures lab link I has code and tation of sk tation of sk tation of sk station of sk a = 0.7) See sample you can ge colab link: p=sharing axis tation of sk station of sk	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	### Training data ### Training data ### Training data ### Sklearn.# ### Train a linear many and a land	preprocess	Sklearn's Lin perficients and perficients and at shows the Normal to the value are coefficient. The coefficient are coefficient are coefficient. The coefficient are	styperdiger of the styperdiger o	ssion (example in the station of the state o	din y-axis (see documen to which feature	tation of sk tation of sk tation of sk tare and bas a = 0.7) d to Avg. S has the big ength of m structures lab link I has code and tation of sk tation of sk tation of sk station of sk a = 0.7) See sample you can ge colab link: p=sharing axis tation of sk station of sk	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	training-data from sklearn.m from sklearn.m from sklearn.m from sklearn.m x = dr olean.s x = dr	preprocess	Sklearn's Lin perficients and perficients and at shows the Normal to the value are coefficient. The coefficient are coefficient are coefficient. The coefficient are	styperdiger of the styperdiger o	ssion (example in the station of the state o	din y-axis (see documen to which feature	tation of sk tation of sk tation of sk tare and bas a = 0.7) d to Avg. S has the big ength of m structures lab link I has code and tation of sk tation of sk tation of sk station of sk a = 0.7) See sample you can ge colab link: p=sharing axis tation of sk station of sk	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	training data from aklearn. from aklearn. from aklearn. from aklearn. from aklearn. from aklearn. generate aklearn. generate aklearn. generate aklearn. generate aklearn. generate aklearn. from aklearn. from aklearn. from aklearn. from sklearn. generate aklearn. generat	proposes of the proposes of th	Sklean"s Lin befficients and at shows the violations to the service officient. at shows the violation at shows the violation of the service officient. The street officient of the service officient officie	squared absolute or y predly y	ssion (example in the st x-axis and y predict x-axis and y predict x-axis and y predict x-axis and y predict agression error ptreg.coefsep 17355707]] least", y = "predict addicted Values st 400	din y-axis (see documen to which feature	mse, r2); d to Avg. S are and bas mse, r2); d to Avg. S has the big ength of m structures lab clink I ha code and tation of sl structures lab clink I ha code and a = 0.7) structures lab clink I ha code and a = 0.7)	elearn's metric ded on that explain the ded on that explain the ded on that explain the ded on the	and the modern and the slide state of the slide slide state of the slide slide state of the slide sl	an App, Time on Ap
	training data from sklearn, from sklearn, from sklearn, from sklearn, gatler lean, actler lean, actler lean, actler lean, actler sklearn, gatler sklearn, from sklearn, f	produces of the composition of t	Sklean's Lin Sklean's Lin Sklean's Lin Defficients and at shows the variet the value at shows the variet mean— aport mean— aport yetst, comparison Test Adoo Ado	squared absorbuted by predly ypredly y	ssion (example in the st x-axis and y predict x-axis and y predict x-axis and y predict x-axis and y predict agression error ptreg.coefsep 17355707]] least", y = "predict addicted Values st 400	cted", alph cted", alph cted", alph cted", alph cted", alph cteds. Also, the training set pression? diny-axis (see documen characteristics, x4 XB_train_T. ".format(train) cted, alph cted,	ion slide/or tation of sl tation of sl tre and bas mse, r2); d to Avg. S has the big ength of m entructures lab link I ha er code and tation of sl er code and a = 0.7) st tation of sl	ession Length gest effect on lead on that ext seed on that ext lead on that ext lead on that ext extern's metric lead on that ext lead on that	Time of the modern and the slide of the slid	an play with eor and equation of the state o
	### A Command	produces of the composition of t	Sklearn's Lin Sklearn's Lin Sklearn's Lin Defficients and at shows the North the value at shows the North the short the s	squared square	ssion (example in the station (example in the egression egression egression egression error egression example into various ould recommend you station (example into various ould recommend you in dealter (example into various example into various ould recommend you in dealter (example into example into various example into various ould recommend you in dealter (example into example into various example into vario	cted", alph cted", alph cted", alph cted", alph cted", alph cteds. Also, the training set pression? diny-axis (see documen characteristics, x4 XB_train_T. ".format(train) cted, alph cted,	ion slide/or tation of sl tation of sl tre and bas mse, r2); d to Avg. S has the big ength of m entructures lab link I ha er code and tation of sl er code and a = 0.7) st tation of sl	ession Length gest effect on lead on that ext seed on that ext lead on that ext lead on that ext extern's metric lead on that ext lead on that	Time of the modern and the slide of the slid	an play with eor and equation of the state o
	training-data from aktean from aktean from sklear generate an from sklear generate an generate an generate an from sklear generate an	detrics in the protect of the protec	Sklearn's Line Skleficients Line Skleficients and selections and shows the viring the value selections and shows the viring the value selections and shows the viring the value selections and the v	st) South and which are the same and which are the same and and are the same and are the s	stin (example in the stin (exa	cted", alph ain, X3 trai win, X3 trai cted", alph	ion slide/or tation of sl tation of sl tre and bas mse, r2); d to Avg. S has the big ength of m entructures lab link I ha er code and tation of sl er code and a = 0.7) st tation of sl	ession Length gest effect on lead on that ext seed on that ext lead on that ext lead on that ext extern's metric lead on that ext lead on that	Time of the modern and the slide of the slid	an play with eor and equation of the state o
	training data from sklearous,	## Company of the com	Sklearis Ling Sk	500 te squared above y pred) y pred, ket on a close to the c	ssion (example in the fit was as and y predicted MSE, RMSE, and R*2 hocefficient belongs agreed to be a second and a second agreed a	inch correspon clinear regress din y-axis (see documen to which featu and the class (the transion) transion? din y-axis (see documen to which featu and the class (the transion) transion? din y-axis (see documen the sklearn transion? the sklearn	See sample you can get and base and base and base and base and base are are are are are are are are are ar	ession Length glack fine state set (title = set (title =	Time of the modern and the slide of the slid	an play with eor and equation of the state o

In [28]: cost list = [] epoch list = [] predicted_list = [] n = 200t0, t1 = 500, 5000 # learning schedule hyperparameters def learning_schedule(t): **return** t0 / (t + t1) theta = np.random.randn(5,1) # random initialization for epoch in range(n_epochs): **if** i==101: i = 0 random_index = np.random.randint(m) #print(random index) xi = XB_train[random_index:random_index+1] yi = y_train[random_index:random_index+1] gradients = 2 * xi.T.dot(xi.dot(theta) - yi) eta = learning_schedule(epoch * m + i) theta = theta - eta * gradients i += 1 y pred = np.dot(theta.T, XB test.T) cost = np.mean(np.square(y_test-y_pred)) # MSE (Mean Squared Error) **if** epoch%10==0: cost list.append(cost) epoch_list.append(epoch) theta array([[5.00120749e+02],Out[28]: [2.60568941e+01], [3.84859364e+01], [-4.92987914e-01], [6.13292936e+01]]) In [29]: plt.figure(figsize = (10,10)) plt.xlabel("epoch") plt.ylabel("cost") plt.plot(epoch_list,cost_list) [<matplotlib.lines.Line2D at 0x24011899700>] Out[29]: 175000 150000 125000 100000 75000 50000 25000 125 175 100 150 epoch In [30]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y test) comparison['predicted'] = pd. Series (y_pred[0,:]) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted")plt.show() Test versus Predicted Values 700 500 400 300 400 500 700 test In [31]: mae = mean absolute error(y test, y pred[0,:]) mse = mean_squared_error(y_test, y_pred[0,:]) rmse = mean_squared_error(y_test, y_pred[0,:], squared = False) r2 = r2_score(y_test, y_pred[0,:]) print("MAE: {}\nMSE: {}\nR_squared: {}".format(mae, mse, rmse, r2)) MAE: 7.281177352718996 MSE: 81.2899398066397 RMSE: 9.016093378322992 R squared: 0.9888011624677553 For larger datasets, it can converge faster as it causes updates to the parameters more frequently. Due to frequent updates, the steps taken towards the minima are very noisy. This can often lean the gradient descent into other directions. 8 SGDRegressor from sklearn 1. Use sklearn's SGDRegressor to train a model for our data set. Put a reasonable iteration and tolerance and learning steps so that we can get coefficients close to normal equation 2. Display the theta values. Are they very close to sklearn's linear regression? 3. Predict for the test data 4. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 5. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) In [32]: from sklearn.linear model import SGDRegressor sgd reg = SGDRegressor(max iter=1000, tol=1e-3, penalty=None, eta0=0.1) sgd_reg.fit(X_train, y_train.ravel()) sgd reg.intercept , sgd reg.coef (array([500.35067437]), Out[32]: array([25.58500836, 38.67331153, -0.09914352, 61.48881844])) They are close to Sklearns linear regression. In [33]: y pred = sgd reg.predict(X test) In [34]: plt.figure(figsize = (10,10)) comparison = pd.DataFrame() comparison['test']=pd.Series(y_test) comparison['predicted']=pd.Series(y pred) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted")plt.show() Test versus Predicted Values 700 600 500 400 300 300 400 500 600 700 In [35]: mae = mean_absolute_error(y_test, y_pred) mse = mean_squared_error(y_test, y_pred) rmse = mean_squared_error(y_test, y_pred, squared = False) r2 = r2 score(y test, y pred) print("MAE: {}\nMSE: {}\nRMSE: {}\nR_squared: {}".format(mae, mse, rmse, r2)) MAE: 7.207159003034535 MSE: 79.19536996875892 RMSE: 8.899178050177383 R squared: 0.9890897190513886 9 Mini-batch Gradient Descent Briefly explain how mini-batch can overcome the limitations of Batch gradient descent and SGD. Easily fits in the memory It is computationally efficient Benefit from vectorization If stuck in local minimums, some noisy steps can lead the way out of them Average of the training samples produces stable error gradients and convergence 10 Polynomial of degree 2 1. Use sklearn's Polynomial features to degree = 2 on our training and test set 2. Use linearRegression on the new polynomial features 3. Predict for test set 4. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 5. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) In [36]: from sklearn.preprocessing import PolynomialFeatures poly = PolynomialFeatures(2) poly X train = poly.fit transform(X train) poly_X_test = poly.fit_transform(X_test) In [37]: reg=LinearRegression() reg.fit(poly X train, y train) LinearRegression() Out[37]: In [38]: y_pred = reg.predict(poly_X_test) In [39]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y_test) comparison['predicted']=pd.Series(y pred[:,0]) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted") plt.show() Test versus Predicted Values 700 600 500 400 300 300 400 500 600 700 In [40]: mae = mean absolute error(y test, y pred[:,0]) mse = mean_squared_error(y_test, y_pred[:,0]) rmse = mean_squared_error(y_test, y_pred[:,0], squared = False) r2 = r2_score(y_test, y_pred[:,0]) print("MAE: {}\nMSE: {}\nR squared: {}".format(mae, mse, rmse, r2)) MAE: 7.432195703827286 MSE: 85.34745505512038 RMSE: 9.238368636026621 R squared: 0.9882421824247091 11 Polynomial of degree 3 1. Use sklearn's Polynomial features to degree = 3 on our training and test set 2. Use linearRegression on the new polynomial features 3. Predict for test set 4. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 5. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) In [41]: poly = PolynomialFeatures(3) poly_X_train = poly.fit_transform(X_train) poly_X_test = poly.fit_transform(X_test) In [42]: reg=LinearRegression() reg.fit(poly X train, y train) LinearRegression() Out[42]: In [43]: y pred = reg.predict(poly X test) In [44]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y_test) comparison['predicted']=pd.Series(y_pred[:,0]) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted")plt.show() Test versus Predicted Values 700 600 400 300 500 300 600 700 test In [45]: mae = mean_absolute_error(y_test, y_pred[:,0]) mse = mean_squared_error(y_test, y_pred[:,0]) rmse = mean_squared_error(y_test, y_pred[:,0], squared = False) r2 = r2_score(y_test, y_pred[:,0]) print("MAE: {}\nMSE: {}\nR squared: {}".format(mae, mse, rmse, r2)) MAE: 7.695801008025619 MSE: 94.96886930852006 RMSE: 9.745197243181897 R squared: 0.9869166966965791 12 Learning Curve 1. Generate learning curve with linearRegression 2. Generate learning curve with polynomial regression with degree = 5 3. Interpret the result In [46]: def plot_learning_curves(model, X, y): X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101) train_errors, pred_errors = [], [] for m in range(1, len(X_train)): model.fit(X_train[:m], y_train[:m]) y_train_predict = model.predict(X_train[:m]) y_test_pred = model.predict(X_test) train_errors.append(mean_squared_error(y_train[:m], y_train_predict)) pred_errors.append(mean_squared_error(y_test , y_test_pred)) plt.plot(np.sqrt(train_errors), "r-+", linewidth = 2, label = "train") plt.plot(np.sqrt(pred_errors), "b-", linewidth = 3, label = "val") print("Train Error: {}\nVal Error: {}".format(train_errors[348], pred_errors[348])) In [47]: lin reg = LinearRegression() plot learning curves(lin reg, X scaled, y) Train Error: 107.11878264719886 Val Error: 79.85787475413264 100 80 60 40 20 0 50 100 150 200 250 300 350 In [48]: poly = PolynomialFeatures(5) poly_X = poly.fit_transform(X_scaled) plot_learning_curves(lin_reg, poly_X, y) Train Error: 73.15139658106436 Val Error: 6395.80957778272 20000 15000 10000 5000 100 150 200 250 50 300 350 The model with data fitted to a 5 feature Polynomial is showing considerable signs of overfitting, with a much higher validation error than training error. 13 Regularization Explain the purpose of regularization For the following Regularization method, use the polynomial degree 3 data set Since the polynomial dataset is prone to overfitting, regularization can help this issue by constraining the weights of the model. 14 Ridge Regression 1. Use sklearn's Ridge to train the data set (use the polynomial degree 3 data set) 2. Predict for test set 3. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 4. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) In [49]: poly = PolynomialFeatures(3) poly_X_train = poly.fit_transform(X train) poly_X_test = poly.fit_transform(X_test) In [50]: from sklearn.linear model import Ridge ridge_reg = Ridge(alpha=1 , solver= "cholesky") ridge reg.fit(poly X train , y train) y_pred= ridge_reg.predict(poly_X_test) In [51]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y_test) comparison['predicted']=pd.Series(y_pred[:,0]) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted") plt.show() Test versus Predicted Values 700 600 500 300 300 400 500 700 600 In [52]: mae = mean_absolute_error(y_test, y_pred[:,0]) mse = mean_squared_error(y_test, y_pred[:,0]) rmse = mean_squared_error(y_test, y_pred[:,0], squared = False) r2 = r2_score(y_test, y_pred[:,0]) print("MAE: {}\nRMSE: {}\nR_squared: {}".format(mae, mse, rmse, r2)) MAE: 7.855335064312291 MSE: 97.6595914733688 RMSE: 9.882286753245364 R squared: 0.9865460117084953 15 SGDRegressor for Ridge 1. Use sklearn's SGDRegressor for Ridge Regression 2. Predict for test set 3. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 4. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) In [53]: from sklearn.linear model import SGDRegressor sgd reg = SGDRegressor(penalty="12") sgd reg.fit(poly X train, y train.ravel()) y pred = sgd reg.predict(poly X test) In [54]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y test) comparison['predicted']=pd.Series(y_pred) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted") plt.show() Test versus Predicted Values 4000 2000 0 -2000-4000 -6000-8000 300 400 500 600 700 In [55]: mae = mean absolute error(y test, y pred) mse = mean squared error(y test, y pred) rmse = mean_squared_error(y_test, y_pred, squared = False) r2 = r2_score(y_test, y_pred) print("MAE: {}\nMSE: {}\nR squared: {}".format(mae, mse, rmse, r2)) MAE: 797.3215945460914 MSE: 2126788.367467365 RMSE: 1458.351249688279 R squared: -291.99514121167556 16 Lasso Regression 1. Use sklearn's Lasso 2. Predict for test set 3. Generate a scatter plot that shows the Y test on the x-axis and y predicted in the y-axis 4. Use sklearn's metrics to print the value of MAE, MSE, RMSE, and R^2 (see documentation of sklearn's metrics) 5. How Lasso perform the regularization and how does that affect the thetas? In [56]: from sklearn.linear model import Lasso lasso reg = Lasso(alpha = 0.1) lasso_reg.fit(poly_X_train, y_train) y pred = lasso reg.predict(poly X test) In [57]: plt.figure(figsize = (10,10)) comparison = pd.DataFrame() comparison['test']=pd.Series(y test) comparison['predicted']=pd.Series(y_pred) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted")plt.show() Test versus Predicted Values 700 600 500 400 300 300 400 500 600 700 In [58]: mae = mean absolute error(y test, y pred) mse = mean squared error(y test, y pred) rmse = mean_squared_error(y_test, y_pred, squared = False) r2 = r2 score(y test, y pred) print("MAE: {}\nMSE: {}\nR squared: {}".format(mae, mse, rmse, r2)) MAE: 7.679627716794209 MSE: 93.21215816943548 RMSE: 9.654644383375054 R squared: 0.987158708471769 17 Elastic Net 1. Use sklearn's ElasticNet 2. Predict for test set 3. Generate a scatter plot that shows the Y test in x axis and y predicted in y axis 4. Use sklearn's metrics to print the value of MAE, MSE, RMSE and R^2 (see documentation of sklearn's metrics) 5. How ElasticNet different compared to Lasso and RIDGE perform the regularization and how does that affect the thetas? In [59]: from sklearn.linear model import ElasticNet elastic net = ElasticNet(alpha=0.1, 11 ratio=0.5) elastic_net.fit(poly_X_train, y_train) y pred = elastic_net.predict(poly_X_test) In [60]: plt.figure(figsize = (10,10))comparison = pd.DataFrame() comparison['test']=pd.Series(y test) comparison['predicted'] = pd. Series (y pred) sns.scatterplot(data = comparison, x = "test", y = "predicted", alpha = 0.7).set(title = "Test versus Predicted")Test versus Predicted Values 800 700 600 predicted 500 400 300 500 700 test In [61]: mae = mean_absolute_error(y_test, y_pred) mse = mean_squared_error(y_test, y_pred) rmse = mean_squared_error(y_test, y_pred, squared = False) r2 = r2_score(y_test, y_pred) print("MAE: {}\nMSE: {}\nRMSE: {}\nRMSE: {}\nRmse, rmse, r2)) MAE: 11.378641625545777 MSE: 261.6917512822405 RMSE: 16.176889419237572 R squared: 0.9639482645317565 Elastic net is a mix of Ridge and Lassos regularization terms with a mix ratio of r. In general, Elastic Net is preferred over Lasso since Lasso may behave erratically when the number of features is greater than the number of training instances or when several features are strongly correlated. 18 Bonus Question: In most of the above cases, for example, LinearRegression of sklearn, (Q4 above), we have used scaled data set for training. However, in a real-life scenario, you would like to predict the yearly amount spent for a new instance. The real data will not be scaled. How would you use the model for this case to predict this instance? [35.49726772511229,12.655651149166752,39.57766801952616,4.082620632952961] = ?\ Write necessary code so that it will predict a reasonable value for the amount spent. This is very close to our first training record. In [62]: reg=LinearRegression() reg.fit(X_train, y_train) X test = scaler.transform(np.expand dims(np.array([35.49726772511229,12.655651149166752,39.57766801952616,4.082 y_pred = reg.predict(X_test)