MA1391

CP_Linear Regression

PRESENT BY GROUP 2

Members in group:

Phan Quoc Trung

Le Huy Hoan

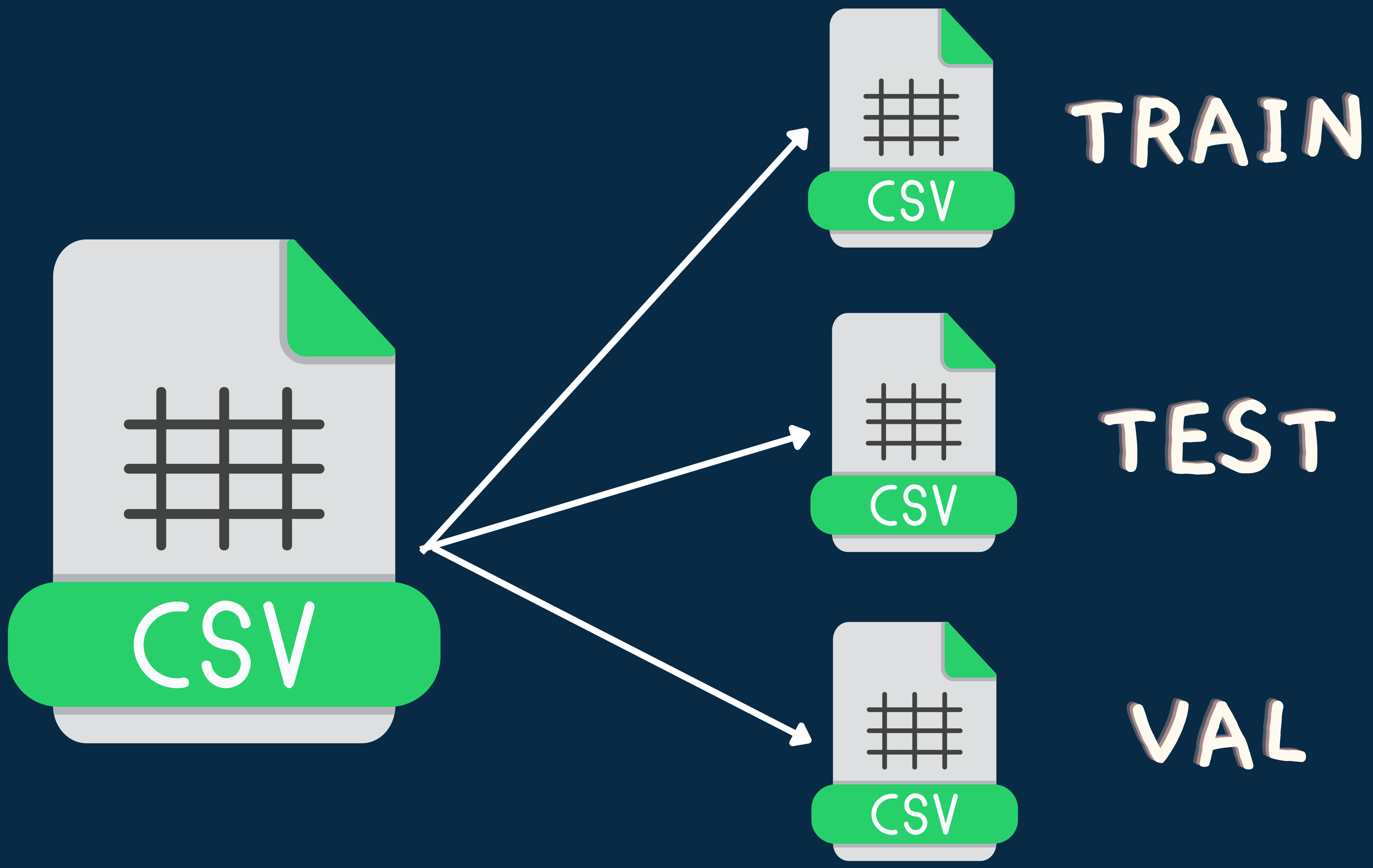
Hoang Thanh Lam

Ngo Dinh An

Table of contents

LINEAR REGRESSION MODEL

TARGET: USING LINEAR REGRESSION MODEL TO PREDICT SEGMENT PHONE PRICE RANGE THROUGH ITS INTERNAL ATTRIBUTES



HTTPS://WWW.KAGGLE.COM/DATASETS/IABHISHEKOFFICIAL/MOBILE-PRICE-CLASSIFICATION?RESOURCE=DOWNLOAD

TRAIN TABLE

CSV

1	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memo	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range
2	842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1	1
3	1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0	2
4	563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0	2
5	615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	0	2
6	1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	1	0	1
7	1859	0	0.5	1	3	0	22	0.7	164	1	7	1004	1654	1067	17	1	10	1	0	0	1
8	1821	0	1.7	0	4	1	10	0.8	139	8	10	381	1018	3220	13	8	18	1	0	1	3
9	1954	0	0.5	1	0	0	24	0.8	187	4	0	512	1149	700	16	3	5	1	1	1	0
10	1445	1	0.5	0	0	0	53	0.7	174	7	14	386	836	1099	17	1	20	1	0	0	0
1990	1547	1	2.9	0	2	0	57	0.4	114	1	3	347	957	1620	9	2	19	0	1	1	1
1991	586	0	2.8	0	2	0	15	0.2	83	3	11	241	854	2592	12	8	3	0	0	0	1
1992	1617	1	2.4	0	8	1	36	0.8	85	1	9	743	1426	296	5	3	7	1	0	0	0
1993	1882	0	2	0	11	1	44	0.8	113	8	19	4	743	3579	19	8	20	1	1	0	3
1994	674	1	2.9	1	1	0	21	0.2	198	3	4	576	1809	1180	6	3	4	1	1	1	0
1995	1467	1	0.5	0	0	0	18	0.6	122	5	0	888	1099	3962	15	11	5	1	1	1	3
1996	858	0	2.2	0	1	0	50	0.1	84	1	2	528	1416	3978	17	16	3	1	1	0	3
1997	794	1	0.5	1	0	1	2	0.8	106	6	14	1222	1890	668	13	4	19	1	1	0	0
1998	1965	1	2.6	1	0	0	39	0.2	187	4	3	915	1965	2032	11	10	16	1	1	1	2
1999	1911	0	0.9	1	1	1	36	0.7	108	8	3	868	1632	3057	9	1	5	1	1	0	3
2000	1512	0	0.9	0	4	1	46	0.1	145	5	5	336	670	869	18	10	19	1	1	1	0

Val

1	id	battery_pc	blue	clock_spee	dual_sim	fc	four_g	int_memo	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_scre	wifi	price_range
2	1	1225	0	0.7	1	6	0	60	0.1	107	2	15	10	1567	2423	17	11	6	1	0	0	2
3	2	1970	1	0.5	1	0	1	15	1	132	2	0	1399	1684	1658	15	9	20	1	1	1	2
4	3	1186	1	0.5	1	2	0	21	0.4	160	8	4	68	584	2361	17	8	7	1	0	0	1
5	4	1762	0	0.7	0	7	0	60	0.1	157	4	10	643	790	1380	14	5	14	1	0	0	1
6	5	1731	1	1.4	1	4	1	4	0.5	163	6	18	809	1988	3892	5	1	4	1	1	1	3
7	6	852	0	1.8	1	5	1	8	0.8	160	5	9	683	1349	315	7	6	20	1	0	0	0
8	7	848	1	1.5	1	4	1	36	0.3	151	4	16	56	527	2289	10	2	19	1	1	1	1
9	8	1575	0	2.5	1	0	1	11	0.2	185	7	20	492	818	2182	7	3	17	1	1	0	2
10	9	1554	0	2.8	1	7	0	23	0.1	105	6	18	699	1492	2184	13	9	20	1	0	1	2
490	489	1547	1	2.9	0	2	0	57	0.4	114	1	3	347	957	1620	9	2	19	0	1	1	1
491	490	586	0	2.8	0	2	0	15	0.2	83	3	11	241	854	2592	12	8	3	0	0	0	1
492	491	1617	1	2.4	0	8	1	36	0.8	85	1	9	743	1426	296	5	3	7	1	0	0	0
493	492	1882	0	2	0	11	1	44	0.8	113	8	19	4	743	3579	19	8	20	1	1	0	3
494	493	674	1	2.9	1	1	0	21	0.2	198	3	4	576	1809	1180	6	3	4	1	1	1	0
495	494	1467	1	0.5	0	0	0	18	0.6	122	5	0	888	1099	3962	15	11	5	1	1	1	3
496	495	858	0	2.2	0	1	0	50	0.1	84	1	2	528	1416	3978	17	16	3	1	1	0	3
497	496	794	1	0.5	1	0	1	2	0.8	106	6	14	1222	1890	668	13	4	19	1	1	0	0
498	497	1965	1	2.6	1	0	0	39	0.2	187	4	3	915	1965	2032	11	10	16	1	1	1	2
499	498	1911	0	0.9	1	1	1	36	0.7	108	8	3	868	1632	3057	9	1	5	1	1	0	3
500	499	1512	0	0.9	0	4	1	46	0.1	145	5	5	336	670	869	18	10	19	1	1	1	0

CSV

TEST

1	id	battery_power	blue	clock_spee	dual_sim	fc	four_g	int_memo	m_dep	mobile_wt	n_cores	pc	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_scre	wifi
2	1	1043	1	1.8	1	14	0	5	0.1	193	3	16	226	1412	3476	12	7	2	0	1	0
3	2	841	1	0.5	1	4	1	61	0.8	191	5	12	746	857	3895	6	0	7	1	0	0
4	3	1807	1	2.8	0	1	0	27	0.9	186	3	4	1270	1366	2396	17	10	10	0	1	1
5	4	1546	0	0.5	1	18	1	25	0.5	96	8	20	295	1752	3893	10	0	7	1	1	0
6	5	1434	0	1.4	0	11	1	49	0.5	108	6	18	749	810	1773	15	8	7	1	0	1
7	6	1464	1	2.9	1	5	1	50	0.8	198	8	9	569	939	3506	10	7	3	1	1	1
8	7	1718	0	2.4	0	1	0	47	1	156	2	3	1283	1374	3873	14	2	10	0	0	0
9	8	833	0	2.4	1	0	0	62	0.8	111	1	2	1312	1880	1495	7	2	18	0	1	1
10	9	1111	1	2.9	1	9	1	25	0.6	101	5	19	556	876	3485	11	9	10	1	1	0
990	989	1653	1	1.3	0	0	0	49	0.7	99	7	4	206	936	377	9	7	10	1	0	0
991	990	635	1	1.9	1	0	1	57	1	136	6	11	1257	1597	3954	15	4	20	1	1	0
992	991	1807	0	1.2	0	4	0	37	0.8	162	1	11	246	932	2741	7	1	9	1	1	1
993	992	1797	1	2.6	0	4	0	42	0.6	174	3	20	57	1169	3359	16	6	18	1	1	1
994	993	1895	0	0.5	1	0	1	62	0.9	99	2	0	1019	1698	2563	10	8	13	1	0	1
995	994	567	1	2.7	1	14	1	56	0.4	165	8	17	555	1290	336	7	6	7	1	1	1
996	995	936	1	1.4	1	0	0	46	0.8	139	2	0	265	886	684	8	5	12	1	1	1
997	996	1700	1	1.9	0	0	1	54	0.5	170	7	17	644	913	2121	14	8	15	1	1	0
998	997	609	0	1.8	1	0	0	13	0.9	186	4	2	1152	1632	1933	8	1	19	0	1	1
999	998	1185	0	1.4	0	1	1	8	0.5	80	1	12	477	825	1223	5	0	14	1	0	0
1000	999	1533	1	0.5	1	0	0	50	0.4	171	2	12	38	832	2509	15	11	6	0	1	0
1001	1000	1270	1	0.5	0	4	1	35	0.1	140	6	19	457	608	2828	9	2	3	1	0	1

Methods

Simple Linear Regression

$$y = \beta_0 + \beta_1 X + \varepsilon$$

Multi Linear Regression

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m$$

$$\hat{y} = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n$$

Methods

$$\hat{y} = h(\theta) = \theta^T x$$

WE COMPUTE A LOSS FUNCTION. THE GOAL OF THE TRAINING PROCESS IS TO FIND THE VALUES OF THETA (Θ) THAT MINIMIZE THE LOSS FUNCTION.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

$$J(\theta_0, \theta_1, \theta_2, \ldots, \theta_m) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

NEXT STEP »

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Now,

$$\frac{\partial}{\partial \Theta} J_\Theta = \frac{\partial}{\partial \Theta} \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y]^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x_i) - y) \frac{\partial}{\partial \Theta_j} (\Theta x_i - y)$$

$$= \frac{1}{m} (h_\Theta(x_i) - y) x_i$$

Therefore,

$$\boxed{\Theta_j := \Theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} [(h_\Theta(x_i) - y) x_i]}$$

Gradient descent algorithm

$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)}$$

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}}$$

}

update θ_0 and θ_1 simultaneously

$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

Deploy in Python

Feel free to get in touch

Contact Google for any questions or clarifications