

INF 395 Final Project Report

STUDENT PERFORMANCE PREDICTION SYSTEM

Sapybek Dinmukhamed-220103053

Aitimbet Omar-220103131

05.05.2025

INTRODUCTION

Here our project contains 2 datasets, 1 database , 4 coding files, then 5 models . Below we will provide the information about each of them . But let me introduce our aim. We want to build a system where we can identify the academic success or failure of the students, by looking at their personal information . If a student can have academic success then what is the probability of success otherwise what is the probability of failure (which is not hard just $1 - P(\text{success})$). Now let's move on to the project itself .

DATESETS

At first we gathered data from SDU University portal oldmy.sdu.edu.kz . This is the first dataset where it is not cleaned , not preprocessed, not wrangled .

Reports (23).xlsx

No	CLASS	PROG_CODE	EDU_LAN	SPECIALTY	DEP_CODE	EDU_CODE	EDU_LEVEL	STATUS	DIR	MAR	GENDER	BIRTH_DATE	AGE	COUNTRY	NATIONAL	CONCISE_ADDRESS	GRANT_C	SCHOOL	ATTTESTAT	GPA	FAULTS	CC	REPEATS
1	3	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		Male	12/15/97	27	Kazakh			SG			1.8	CSS 102,CS	0	
2	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		67 Male	12/15/97	27	Kazakh			SG			0.17	CSS 101,IE	0	
3	2	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		Male						SG	School		1.24	CSS 102,IE	0	
4	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		91 Male	11/15/96	28	Uzbek			SG	Lycee (KTL)		1.44		0	
5	3	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying		93 Male	12/28/92	31	Kazakhstan			SG	Lycee (KTL)		1.48	CSS 216,CS	0	
6	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying		0 Male	1/1/96	28	Turk			SG			1.86	INF 103,M	0	
7	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying		100 Male	7/5/97	27	Kazakh			SG			0.86	INF 103,M	0	
8	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying		92 Female	9/28/95	29				SG			3.08		0	
9	4	10106	KZ	Көмөкчү	DEP_CODE	COMF	ENG	B	Graduated		86 Male	9/15/94	30	Kazakhstan		Казакстан, Алармат облысы(06), Алармат уези	SG	School		2.59		0	
10	1	10106	KZ	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		82 Female	6/28/98	25	Kazakh			SG			0.94	INF 102,IN	0	
11	2	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		91 Male	5/29/97	27	Kazakh			SG	NIS		1.6	CSS 201,CS	0	
12	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying		108 Male	3/4/97	27	Kazakh			SG			0	HSS 111,IE	0	
13	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying		88 Male	11/15/96	28	Kazakh			SG			0.68	CSS 102,CS	0	
14	3	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying		90 Female	11/23/94	30	Kazakhstan			SG			1.19	CSS 102,CS	0	
15	1	10106	KZ	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		73 Female	5/15/95	27	Kazakhstan		Казакстан, Мангыстау облысы(12), Акты уези	SG	School	4.5	1.86	INF 152,M	0	
16	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying		75 Male	6/20/97	27	Kazakh			SG			0.4	CSS 102,CS	0	
17	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		100 Male	2/23/97	27	Kazakh			SG	Lycee (KTL)		2.25	TUR 101	0	
18	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying		94 Male	3/25/96	28	Kazakh			SG			0.69	CSS 102,CS	0	
19	1	10106	KZ	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying		82 Male	11/1/97	27	Kazakhstan		Казакстан, Кызылорда облысы(11), Кызылорда уези	SG	Lycee	4	0.09	HSS 111,IE	0	
20	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying		62 Male	5/19/96	28	Kazakh			SG			1.05	CSS 102,CS	0	
21	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		75 Male	1/1/94	30	Kazakhstan		Казакстан, Шыгыс Казакстан облысы(16), Шомен каласы	SG	Lycee (KTL)		2.54		0	
22	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		77 Male	4/20/95	29	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		2.47		0	
23	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		100 Female	9/25/95	29	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		2.71		0	
24	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		87 Female	9/1/94	30	Kazakhstan		Казакстан, Павлодар облысы(14), Павлодар каласы	SG	Lycee (KTL)		3.56		0	
25	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		77 Male	6/2/95	29	Kazakhstan		Казакстан, Алармат облысы(06), Алармат каласы	SG	School		3.32		0	
26	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		93 Male	4/17/95	29	Kazakhstan		Казакстан, Актебе облысы(04)	SG	School		3		0	
27	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		88 Male	1/2/96	28	Kazakhstan		Казакстан, Актебе облысы(04), Актебе каласы	SG	School		3.68		0	
28	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		73 Female	2/9/95	29	Kazakhstan		Казакстан, Жамбыл облысы(08), Тараз каласы	UG	Lycee (KTL)		2.84		0	
29	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		88 Female	11/8/94	30	Kazakhstan		Казакстан, Алармат облысы(06)	SG	Daryn		2.89		0	
30	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		77 Male	8/21/94	30	Kazakhstan		Казакстан, Түркістан облысы(13)	SG	Daryn		3.13		0	
31	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		60 Male	1/11/95	29	Kazakhstan		Казакстан, Жамбыл облысы(08), Тараз каласы	SG	Lycee (KTL)		2.82		0	
32	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		68 Male	2/20/94	30	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		2.14		0	
33	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		100 Female	7/1/94	30	Kazakhstan		Казакстан, Акмолта каласы	SG	Lycee (KTL)		2.66		0	
34	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		94 Female	8/29/94	30	Kazakhstan		Казакстан, Алармат облысы(06), Алармат каласы	SG	School		3.1		0	
35	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		70 Female	6/27/95	29	Kazakhstan		Казакстан, Алармат облысы(06), Алармат каласы	SG	School		2.62		0	
36	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		84 Male	3/21/94	30	Kazakhstan		Казакстан, Акмолта каласы	SG	Lycee		3.42		0	
37	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		78 Female	8/23/94	30	Kazakhstan		Казакстан, Акмолта каласы	SG	School		3.34		0	
38	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		85 Female	9/6/94	30	Kazakhstan		Казакстан, Кызылорда облысы(11), Кызылорда каласы	SG	School		3.51		0	
39	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		79 Female	10/24/92	32	Kazakhstan		Казакстан, Жамбыл облысы(08), Тараз каласы	SG	Lycee (KTL)		3.48		0	
40	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		73 Female	2/17/95	29	Kazakhstan		Казакстан, Караганда облысы(09), Жезказган каласы	SG	Gymnazy		2.78		0	
41	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		80 Female	6/27/94	30	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		3.32		0	
42	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		66 Female	2/6/95	29	Kazakhstan		Казакстан, Астана каласы	SG	Lycee (KTL)		2.81		0	
43	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		60 Male	1/27/95	29	Kazakhstan		Казакстан, Акмолта облысы(05), Жезказган уези	UG	School	4	2.58		0	
44	4	10101	EN	Манмат	DEP_CODE	MATH	ENG	B	Graduated		72 Female	4/17/95	29	Kazakhstan		Казакстан, Жамбыл облысы(08)	SG	Lycee (KTL)	5	2.72		0	
45	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		66 Male	2/23/94	30	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		3.77		0	
46	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		70 Female	2/8/94	30	Kazakhstan		Казакстан, Акмолта облысы(05)	SG	School		3.09		0	
47	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		60 Female	3/27/94	30	Kazakhstan		Казакстан, Акмолта облысы(05), Карасай уези, Карасөңк каласы	SG	School	4.5	3.01		0	
48	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		76 Female	1/15/95	29	Kazakhstan		Казакстан, Шымкент каласы	SG	School		3.33		0	
49	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		62 Female	2/7/95	28	Kazakhstan		Казакстан, Мангыстау облысы(12), Мангыт уези	SG	School		2.61		0	
50	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		65 Male	9/2/94	30	Kazakhstan		Казакстан, Жамбыл облысы(08), Тараз уези, Кырты каласы	SG	School		2.61		0	
51	4	10108	KZ	Манмат	DEP_CODE	MATH	ENG	B	Graduated		81 Female	4/17/95	29	Kazakhstan		Казакстан	SG	School		3.24		0	
52	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated		81 Male	7/22/99	25	Kazakhstan		Казакстан, Алармат облысы(06), Талас уези, Кызыр каласы	SG	School		2.46		0	
53	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated		93 Male	2/8/92	32	Kazakhstan		Казакстан, Акмолта облысы(05), Талдыарал каласы	SG	Lycee (KTL)		2.28		0	

Then by the preprocessing we got new preprocessed and cleaned from NA values dataset like

The_Final_dataset.xlsx

Unnamed: 0	No	CLASS	PROG_CODE	EDU_LAN	SPECIALTY	DEP_CODE	EDU_CODE	EDU_LEVEL	STATUS	DIR	MAR	GENDER	BIRTH_DATE	AGE	COUNTRY	NATIONALITY	ADDRESS	CATEGORY	TYPE	TESTAT	GR	GPA	LED	COURSE	REPEATS	REG_DATE
0	0	1	3	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	0 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	1.8	CSS 102,CS	0		
1	1	3	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	67 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	0.17	CSS 101,IE	0		
2	2	3	2	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	0 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	1.24	CSS 102,CS	0		
3	3	4	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	91 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	Lycee (KTL)		0	3.44	Not Failed	0		
4	4	5	3	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying	93 Male	YYYY-00-00	31	Kazakhstan		Not Provided	SG	Lycee (KTL)		0	1.48	CSS 216,CS	0		
5	5	6	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying	0 Male	YYYY-00-00	28	Turkey		Turk	SG	School		0	0	Not Failed	0		
6	6	7	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying	100 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	0.86	INF 103,M	0		
7	7	8	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	92 Female	YYYY-00-00	29	Kazakhstan		Not Provided	SG	School		0	3.08	Not Failed	0		
8	8	9	4	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Graduated	86 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.59	Not Failed	0		
9	9	10	1	10106	KZ	Алармат	DEP_CODE	INF	ENG	B	Not studying	82 Female	YYYY-00-00	25	Kazakhstan		Not Provided	SG	School		0	0.94	INF 102,IN	0		
10	10	11	2	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	91 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	SNS		0	1.6	CSS 201,CS	0		
11	11	12	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying	108 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	0.69	CSS 102,CS	0		
12	12	13	1	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Not studying	88 Male	YYYY-00-00	28	Kazakhstan		Not Provided	SG	School		0	0.68	CSS 102,CS	0		
13	13	14	3	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	90 Female	YYYY-00-00	29	Kazakhstan		Not Provided	SG	School		0	3.05	Not Failed	0		
14	14	15	1	10106	KZ	Алармат	DEP_CODE	INF	ENG	B	Not studying	73 Female	YYYY-00-00	23	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		4.5	1.86	INF 152,M	0		
15	15	16	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	75 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	0.4	CSS 102,CS	0		
16	16	17	1	10107	EN	Көмөкчү	DEP_CODE	COMF	ENG	B	Not studying	100 Male	YYYY-00-00	27	Kazakhstan		Not Provided	SG	School		0	2.57	Not Failed	0		
17	17	18	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	94 Male	YYYY-00-00	28	Kazakhstan		Not Provided	SG	School		0	0.69	CSS 102,CS	0		
18	18	19	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	62 Male	YYYY-00-00	28	Kazakhstan		Not Provided	SG	School		0	1.05	CSS 102,CS	0		
19	19	20	1	10104	KZ	Есемеу	DEP_CODE	COMF	ENG	B	Not studying	62 Male	YYYY-00-00	28	Kazakhstan		Not Provided	SG	School		0	1.05	CSS 102,CS	0		
20	20	21	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	75 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.54	Not Failed	0		
21	21	22	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	77 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.54	Not Failed	0		
22	22	23	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	100 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	2.71	Not Failed	0		
23	23	24	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	87 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.71	Not Failed	0		
24	24	25	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	77 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.32	Not Failed	0		
25	25	26	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	93 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	3	Not Failed	0		
26	26	27	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	88 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	3.68	Not Failed	0		
27	27	28	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	73 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan, US	SG	Lycee (KTL)		0	2.84	Not Failed	0		
28	28	29	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	88 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	2.84	Not Failed	0		
29	29	30	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	77 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Daryn		0	3.13	Not Failed	0		
30	30	31	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	88 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.82	Not Failed	0		
31	31	32	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	68 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	1.4	Not Failed	0		
32	32	33	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	100 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.66	Not Failed	0		
33	33	34	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	94 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.66	Not Failed	0		
34	34	35	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	70 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		0	2.62	Not Failed	0		
35	35	36	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	84 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee		0	3.42	Not Failed	0		
36	36	37	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	78 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	3.4	Not Failed	0		
37	37	38	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	85 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	3.51	Not Failed	0		
38	38	39	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	79 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	3.51	Not Failed	0		
39	39	40	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	73 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Gimnazyi		0	2.78	Not Failed	0		
40	40	41	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	80 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	3.32	Not Failed	0		
41	41	42	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	78 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	3.1	Not Failed	0		
42	42	43	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	60 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		0	2.58	Not Failed	0		
43	43	44	4	10101	EN	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	66 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	2.72	Not Failed	0		
44	44	45	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	66 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan, Uyghur	SG	School		0	3.77	Not Failed	0		
45	45	46	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	70 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	3.03	Not Failed	0		
46	46	47	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	60 Female	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		4.5	0.21	Not Failed	0		
47	47	48	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	76 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		0	3.33	Not Failed	0		
48	48	49	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	65 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	3.5	Not Failed	0		
49	49	50	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	65 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.61	Not Failed	0		
50	50	51	4	10108	KZ	Мамекени	DEP_CODE	MATH	ENG	B	Graduated	81 Female	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	School		0	3.24	Not Failed	0		
51	51	52	4	10108	KZ	Алармат	DEP_CODE	INF	ENG	B	Graduated	81 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	4.6	Not Failed	0		
52	52	53	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	93 Male	YYYY-00-00	32	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.28	Not Failed	0		
53	53	54	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	91 Male	YYYY-00-00	31	Kazakhstan		Kazakhstan, Chechen	SG	School		0	2.28	Not Failed	0		
54	54	55	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	97 Male	YYYY-00-00	32	Kazakhstan		Russian	SG	Kazakhstan, payer		0	2.3	Not Failed	0		
55	55	56	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	60 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		0	2.36	Not Failed	0		
56	56	57	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	78 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.4	Not Failed	0		
57	57	58	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	97 Male	YYYY-00-00	31	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.29	Not Failed	0		
58	58	59	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	67 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.9	Not Failed	0		
59	59	60	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	51 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.77	Not Failed	0		
60	60	61	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	83 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	3.2	Not Failed	0		
61	61	62	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	57 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.78	Not Failed	0		
62	62	63	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	74 Male	YYYY-00-00	29	Kazakhstan		Kazakhstan	SG	Lycee (KTL)		0	2.33	Not Failed	0		
63	63	64	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	63 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	School		0	2.7	Not Failed	0		
64	64	65	4	10103	EN	Алармат	DEP_CODE	INF	ENG	B	Graduated	57 Male	YYYY-00-00	30	Kazakhstan		Kazakhstan	SG	Kazakhstan, payer		0	2.8	Not Failed	0		
65	65																									

In the final dataset we also deleted some columns , to focus on the purpose of the project, and do some models with it .

CODING PART

Here in coding part we have 3 sections

- 1) Data wrangling and preprocessing
- 2) Creating models
- 3) Dashboard

SECTION-1

In this section we have a file Final-3.ipynb . Where we did all the wrangling here we also messed up so we do not comment this part cause here all the stuff is understandable.

1)We begin by importing necessary libraries for data manipulation, visualization, and date handling

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import re

import openpyxl

from datetime import datetime
```

2)This block of code cleans and standardizes the **CONCISE_ADDRESS** column:

```
df['CONCISE_ADDRESS'] = df['CONCISE_ADDRESS'].apply(lambda x: ', '.join(x.split(', ')[:-2])
if isinstance(x, str) else x)

df['CONCISE_ADDRESS'] = df['CONCISE_ADDRESS'].fillna('Not Provided')
```

3)The code fills missing values in the GRANT_CATEGORY column:

```
df['GRANT_CATEGORY'] = df['GRANT_CATEGORY'].fillna('payer')
```

4)Here, the code calculates the missing values for AGE and BIRTH_DATE using the current date and the median age:

```
df['BIRTH_DATE'] = pd.to_datetime(df['BIRTH_DATE'])
```

```
today = datetime.today()
```

```
median_age = int(df['AGE'].median())
```

5)The following code fills the AGE column if it's missing based on the BIRTH_DATE:

```
def fill_age(row):
```

```
    if pd.isna(row['AGE']) and pd.notna(row['BIRTH_DATE']):
```

```
        return today.year - row['BIRTH_DATE'].year - (
```

```
            (today.month, today.day) < (row['BIRTH_DATE'].month, row['BIRTH_DATE'].day)
```

```
        )
```

```
    elif pd.isna(row['AGE']):
```

```
        return median_age
```

```
    else:
```

```
        return int(row['AGE'])
```

```
df['AGE'] = df.apply(fill_age, axis=1)
```

6)Similarly, missing BIRTH_DATE values are estimated based on the AGE:

```
median_birth_date = pd.to_datetime(df['BIRTH_DATE']).median()
```

```

def fill_birth_date(row):
    if pd.isna(row['BIRTH_DATE']) and pd.notna(row['AGE']):
        age_years = int(row['AGE'])
        estimated_date = datetime(today.year - age_years, today.month, today.day)
        if estimated_date > today:
            estimated_date = datetime(today.year - age_years - 1, today.month, today.day)
        return estimated_date
    elif pd.isna(row['BIRTH_DATE']):
        return median_birth_date
    else:
        return row['BIRTH_DATE']
df['BIRTH_DATE'] = df.apply(fill_birth_date, axis=1)

```

7) This code fills the missing **NATIONALITY values based on the country:**

```

def fill_nationality(row):
    if row['COUNTRY'] == 'Tajikistan' and pd.isna(row['NATIONALITY']):
        return 'Tajik'
    return row['NATIONALITY']

df['NATIONALITY'] = df.apply(fill_nationality, axis=1)
df['NATIONALITY'].fillna('Kazakh', inplace=True)

```

8) Missing **COUNTRY values are filled based on the **NATIONALITY**:**

```
def fill_country(row):
    if pd.isna(row['COUNTRY']):
        if row['NATIONALITY'] == 'Turk':
            return 'Turkey'
        elif row['NATIONALITY'] == 'Uzbek':
            return 'Kazakhstan'
        elif row['NATIONALITY'] == 'Kurd':
            return 'Kazakhstan'
        else:
            return 'Kazakhstan'
    return row['COUNTRY']
df['COUNTRY'] = df.apply(fill_country, axis=1)
```

9)The following code fills missing values in **JOIN_MARKS** and **FAILED_COURSES** columns:

```
df['JOIN_MARKS'] = df['JOIN_MARKS'].fillna(0)
df['FAILED_COURSES'] = df['FAILED_COURSES'].fillna('Not Failed:')
```

10)The **GPA** column is filled with 0 where missing:

```
df['GPA'] = df['GPA'].fillna(0)
```

11)The **ATTESTAT_GPA** column is converted based on specific criteria:

```
def convert_gpa(gpa):
    if pd.isna(gpa):
```

```

    return 0

if gpa > 5 and gpa <= 100:

    return (gpa * 5) / 100

elif gpa > 100:

    return 0

else:

    return gpa

```

```
df['ATTESTAT_GPA'] = df['ATTESTAT_GPA'].apply(convert_gpa)
```

12)After all the processing, we check for any remaining missing values:

```
df.isnull().sum()
```

```

51:

```

	0
Unnamed: 0	0
Nº	0
CLASS	0
PROG_CODE	0
EDU_LANG	0
SPECIALITY	0
DEP_CODE	0
DEP_CODE_F	0
EDU_LEVEL	0
STATUS	0
JOIN_MARKS	0
GENDER	0
BIRTH_DATE	0
AGE	0
COUNTRY	0
NATIONALITY	0
CONCISE_ADDRESS	0
GRANT_CATEGORY	0
SCHOOL_TYPE	0
ATTESTAT_GPA	0
GPA	0
FAILED_COURSES	0
REPEATSYEAR	0

SECTOIN-2

Here in this section we got coding that works with clean from noise data . And here we worked with models like Decision_Tree_Model , Logistic_Regression and Random_Forest all this models we used to define academic success or fail of students .

1)The first step is to import all the required libraries for data manipulation, visualization, model training, and evaluation.

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report
```

2)The dataset is loaded from an Excel file using **pandas.**

```
df = pd.read_excel('.venv/data/The_Final_Dataset.xlsx')
```

3)To get a quick overview of the dataset, the **head() method is used:**

```
df.head()
```

4)We use the **info(), **describe()**, **isnull().sum()** methods to get information about the dataset, such as the number of entries, data types, and missing values:**


```
df.info()

df.describe()

df.isnull().sum()
```

5)Some columns, such as **'Unnamed', 'REG_DATE'**, and other non-essential columns, are dropped from the dataset to simplify the analysis:

```
columns_to_drop = ['Unnamed: 0', 'Unnamed: 0.1', '№', 'REG_DATE', 'CONCISE_ADDRESS',
'EDU_LANG', 'DEP_CODE', 'DEP_CODE_F', 'PROG_CODE', 'BIRTH_DATE', 'REPEATSYEAR',
'EDU_LEVEL']
```

```
df_cleaned = df.drop(columns=columns_to_drop)
```

6)Next, we filter the data to only include students aged 24 or younger:

```
df_cleaned = df_cleaned[df_cleaned['AGE'] <= 24]
```

7)We create a binary target variable **Academic_Success**, where success is defined as a **GPA >= 2.0**:

```
df_cleaned['Academic_Success'] = (df_cleaned['GPA'] >= 2.0).astype(int)
```

8)Categorical variables like **GENDER, STATUS**, and **SCHOOL_TYPE** are encoded into numeric values using **LabelEncoder**:

```
le = LabelEncoder()

df_cleaned['GENDER'] = le.fit_transform(df_cleaned['GENDER'])

df_cleaned['STATUS'] = le.fit_transform(df_cleaned['STATUS'])

df_cleaned['SCHOOL_TYPE'] = le.fit_transform(df_cleaned['SCHOOL_TYPE'])
```

9)Columns with multiple categories (like **SPECIALITY, COUNTRY**, and **NATIONALITY**)

are one-hot encoded:

```
df_cleaned = pd.get_dummies(df_cleaned, columns=['SPECIALITY', 'COUNTRY',  
'NATIONALITY', 'GRANT_CATEGORY'], drop_first=True)
```

10)We handle the **FAILED_COURSES feature by counting the number of failed courses for each student:**

```
def count_failed_courses(value):  
    if pd.isna(value):  
        return 0  
    value = str(value).strip()  
    if value == 'Not Failed:':  
        return 0  
    if ',' in value:  
        return value.count(',') + 1  
    return 1  
  
df_cleaned['FAILED_COUNT'] =  
df_cleaned['FAILED_COURSES'].apply(count_failed_courses)
```

11)We separate the features (X**) and target variable (**y**) for model training:**

```
X = df_cleaned.drop(columns=['GPA', 'FAILED_COURSES', 'Academic_Success'])  
y = df_cleaned['Academic_Success']
```

12)We split the data into training and testing sets:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

13)We scale the features using **StandardScaler:**

```
scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

14)We initialize the machine learning models:

```
dt_model = DecisionTreeClassifier(random_state=42)

lr_model = LogisticRegression(max_iter=1000, random_state=42)

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
```

15)The models are trained using the scaled training data:

```
dt_model.fit(X_train_scaled, y_train)

lr_model.fit(X_train_scaled, y_train)

rf_model.fit(X_train_scaled, y_train)
```

16)We make predictions on the test data:

```
dt_predictions = dt_model.predict(X_test_scaled)

lr_predictions = lr_model.predict(X_test_scaled)

rf_predictions = rf_model.predict(X_test_scaled)
```

17)The accuracy of the models is evaluated using **accuracy_score:**

```
dt_accuracy = accuracy_score(y_test, dt_predictions)

lr_accuracy = accuracy_score(y_test, lr_predictions)

rf_accuracy = accuracy_score(y_test, rf_predictions)
```

18)The classification reports provide detailed performance metrics for each model:

```
print("\nDecision Tree Classification Report:")

print(classification_report(y_test, dt_predictions))


print("\nLogistic Regression Classification Report:")

print(classification_report(y_test, lr_predictions))


print("\nRandom Forest Classification Report:")

print(classification_report(y_test, rf_predictions))
```

19)Finally, we predict the academic success of a new student input:

```
input_data = pd.DataFrame({

    'CLASS': [1],

    'GENDER': [1],

    'AGE': [20],

    'STATUS': [0],

    'SCHOOL_TYPE': [0],

    'ATTESTAT_GPA': [2.5],

    'COUNTRY_Kazakhstan': [1],

    'NATIONALITY_Kazakh': [1],

})


input_data = input_data.reindex(columns=X.columns, fill_value=0)

input_data_scaled = scaler.transform(input_data)
```

```
prediction_dt = dt_model.predict(input_data_scaled)
prediction_lr = lr_model.predict(input_data_scaled)
prediction_rf = rf_model.predict(input_data_scaled)
```

20)The predictions are outputted for each model:

```
if prediction_dt[0] == 1:
    print("Decision Tree: The student is likely to succeed academically!")
else:
    print("Decision Tree: The student is likely to fail academically!")

if prediction_lr[0] == 1:
    print("Logistic Regression: The student is likely to succeed academically!")
else:
    print("Logistic Regression: The student is likely to fail academically!")

if prediction_rf[0] == 1:
    print("Random Forest: The student is likely to succeed academically!")
else:
    print("Random Forest: The student is likely to fail academically!")
```

21)The trained models and scaler are saved for later use:

```
import joblib
```

```
joblib.dump(dt_model, 'decision_tree_model.joblib')  
  
joblib.dump(lr_model, 'logistic_regression_model.joblib')  
  
joblib.dump(rf_model, 'random_forest_model.joblib')  
  
joblib.dump(scaler, 'scaler.joblib')
```

SECTION-3

Explanation of streamlit codes

Firstly, **streamlitApp.py**

1)The first step is to import all the necessary libraries and packages that will be used in the application.

```
import streamlit as st  
  
import joblib  
  
import numpy as np  
  
import pandas as pd  
  
from sklearn.preprocessing import StandardScaler
```

2)Here, we load the pre-trained models and scaler from the disk using `joblib.load()`.

```
dt_model = joblib.load('.venv/Trained models/decision_tree_model.joblib')  
  
lr_model = joblib.load('.venv/Trained models/logistic_regression_model.joblib')  
  
rf_model = joblib.load('.venv/Trained models/random_forest_model.joblib')  
  
scaler = joblib.load('.venv/Trained models/scaler.joblib')  
  
columns = joblib.load('.venv/Trained models/columns.joblib')
```

3)Here, we set the title and header of the app using Streamlit.

```
st.title('Student Academic Success Prediction')  
  
st.header("Enter student details for prediction")
```

4)Here, we define various user input fields using Streamlit widgets, where the user can enter details about the student.

```
class_input = st.selectbox("Class", [1, 2, 3])  
  
gender_input = st.radio("Gender", ['Male', 'Female'])  
  
age_input = st.slider("Age", min_value=18, max_value=30, value=20)  
  
status_input = st.selectbox("Status", ['Studying', 'Not studying', 'Graduated'])  
  
school_type_input = st.selectbox("School Type", ['School', 'Lycee (KTL)', 'Lyceum'])  
  
gpa_input = st.slider("GPA", 0.0, 4.0, 2.0)  
  
attestat_gpa_input = st.slider("Attestat GPA", 0.0, 5.0, 2.5)  
  
failed_courses_input = st.text_input("Failed Courses", 'Not Failed:~')  
  
country_input = st.text_input("Country", 'Kazakhstan')  
  
nationality_input = st.text_input("Nationality", 'Kazakh')  
  
grant_input = st.selectbox("Grant Category", ['SG', 'payer'])
```

5)This section converts categorical inputs (e.g., **Gender, **Status**, **School Type**) to numeric values since most machine learning models require numeric inputs.**

```
gender = 0 if gender_input == 'Male' else 1  
  
status = {'Studying': 0, 'Not studying': 1, 'Graduated': 2}[status_input]  
  
school_type = {'School': 0, 'Lycee (KTL)': 1, 'Lyceum': 1}[school_type_input]
```

failed_courses = 1 if failed_courses_input != 'Not Failed:)' else 0

country = 1 if country_input == 'Kazakhstan' else 0

nationality = 1 if nationality_input == 'Kazakh' else 0

grant = 1 if grant_input == 'SG' else 0

6)Here, we create a pandas DataFrame to store the user's inputs in a structured format, ensuring that the model receives the data correctly.

```
input_data = pd.DataFrame({  
    'CLASS': [class_input],  
    'GENDER': [gender],  
    'AGE': [age_input],  
    'STATUS': [status],  
    'SCHOOL_TYPE': [school_type],  
    'GPA': [gpa_input],  
    'ATTESTAT_GPA': [attestat_gpa_input],  
    'FAILED_COURSES': [failed_courses],  
    'COUNTRY_Kazakhstan': [country],  
    'NATIONALITY_Kazakh': [nationality],  
    'GRANT_CATEGORY_SG': [grant]  
})
```

7)This ensures that the input data has the same structure (columns) as the training data by filling any missing columns with 0.

```
input_data = input_data.reindex(columns=columns, fill_value=0)
```


8)The input data is scaled using the same scaler that was applied during training. This helps normalize the input data so the model receives it in the same format it was trained on.

```
input_data_scaled = scaler.transform(input_data)
```

9)The **Predict Academic Success button is used to trigger the prediction process.**

```
predict_button = st.button('Predict Academic Success')
```

10)Once the button is pressed, the models make predictions on the input data.

```
if predict_button:
```

```
    prediction_dt = dt_model.predict(input_data_scaled)
```

```
    prediction_lr = lr_model.predict(input_data_scaled)
```

```
    prediction_rf = rf_model.predict(input_data_scaled)
```

```
    prob_lr = lr_model.predict_proba(input_data_scaled)[0][1] # Probability of success for  
    Logistic Regression
```

```
    prob_rf = rf_model.predict_proba(input_data_scaled)[0][1] # Probability of success for  
    Random Forest
```

11)Finally, the predictions and probabilities of success are displayed for each model.

```
st.write(
```

```
    f"""Decision Tree**: The student is likely to {'succeed' if prediction_dt[0] == 1 else 'fail'}  
    academically!"
```

```
st.write(f"""Probability of Success**: {prob_rf * 100:.2f}%")
```

```
st.write(
```

```
f"""Logistic Regression**: The student is likely to {'succeed' if prediction_lr[0] == 1 else 'fail'} academically!")
```

```
st.write(f"""Probability of Success**: {prob_lr * 100:.2f}%")
```

```
st.write(
```

```
f"""Random Forest**: The student is likely to {'succeed' if prediction_rf[0] == 1 else 'fail'} academically!")
```

```
st.write(f"""Probability of Success**: {prob_rf * 100:.2f}%")
```

Secondly, in the file `streamlitAppSQLite.py` we applied the SQLite and most of the code same like in `streamlitApp.py` but have some changes like with the Database .

```
def init_db():
```

```
    conn = sqlite3.connect('student_data.db')
```

```
    cursor = conn.cursor()
```

```
    # Create the students table if it doesn't exist
```

```
    cursor.execute("""CREATE TABLE IF NOT EXISTS students (
```

```
        id INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
        class INTEGER,
```

```
        gender INTEGER,
```

```
        age INTEGER,
```

```
        status INTEGER,
```

```
        school_type INTEGER,
```

```
        gpa REAL,
```

```
    attestat_gpa REAL,  
    failed_courses INTEGER,  
    country TEXT,  
    nationality TEXT,  
    grant_category TEXT)''')
```

```
# Create the predictions table if it doesn't exist
```

```
cursor.execute('''CREATE TABLE IF NOT EXISTS predictions (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    student_id INTEGER,  
    decision_tree_prediction INTEGER,  
    decision_tree_probability REAL,  
    logistic_regression_prediction INTEGER,  
    logistic_regression_probability REAL,  
    random_forest_prediction INTEGER,  
    random_forest_probability REAL,  
    FOREIGN KEY(student_id) REFERENCES students(id)''')
```

```
conn.commit()
```

```
conn.close()
```

```
# Initialize the database and tables
```

```
init_db()
```

Like here we initialize and create a DataBase , and use it after this .

THE APPLICATION

If you want to run the streamlit application you can use following links:

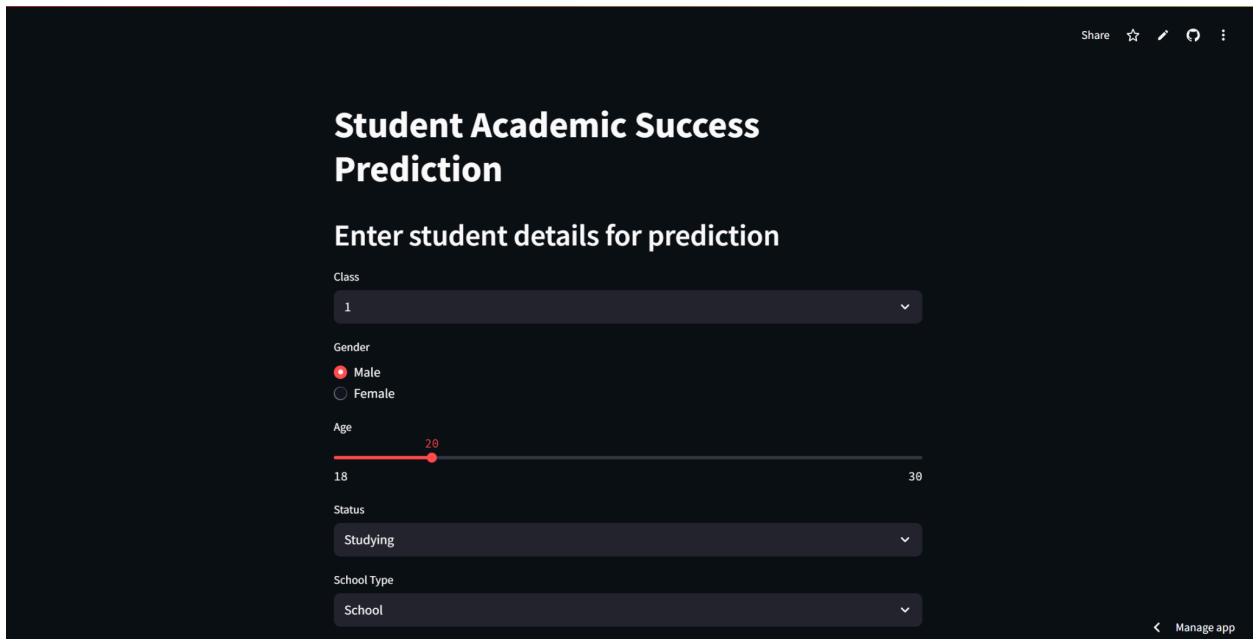
<https://studentacademicsuccesssystem14-p-nosqlite.streamlit.app/>

Or

With SqlLite

<https://studentacademicsuccesssystem14-p-nosqlite.streamlit.app/>

Then you will have the app like in below



The screenshot shows a web application titled "Student Academic Success Prediction". The interface is dark-themed and includes a header with "Share", "star", "edit", "refresh", and "menu" icons. The main heading is "Student Academic Success Prediction". Below it is the instruction "Enter student details for prediction". The form contains the following fields:

- Class:** A dropdown menu with the value "1" selected.
- Gender:** Radio buttons for "Male" (selected) and "Female".
- Age:** A range slider with a minimum of 18, a maximum of 30, and a current value of 20.
- Status:** A dropdown menu with the value "Studying" selected.
- School Type:** A dropdown menu with the value "School" selected.

In the bottom right corner, there is a link that says "< Manage app".

Share
☆
✎
🔄
⋮

School Type
School

GPA
2.00
0.00 4.00

Attestat GPA
2.50
0.00 5.00

Failed Courses
Not Failed:)

Country
Kazakhstan

Nationality
Kazakh

Grant Category
SG

Predict Academic Success

Manage app

Country
Kazakhstan

Nationality
Kazakh

Grant Category
SG

Predict Academic Success

Decision Tree: The student is likely to fail academically!
Probability of Success: 58.62%

Logistic Regression: The student is likely to fail academically!
Probability of Success: 3.23%

Random Forest: The student is likely to succeed academically!
Probability of Success: 58.62%

RESULTS

Once training is complete, the system outputs:

Model Accuracy: A comparison of each model's accuracy.

Classification Reports: Detailed metrics for each model (precision, recall, F1-score).

User Prediction: The system predicts whether a student is likely to succeed or fail academically based on input data.

Here are the performance results of the models:

Decision Tree Accuracy: 88.45%

Logistic Regression Accuracy: 88.88%

Random Forest Accuracy: 89.57%

The evaluation also includes detailed classification reports for each model:

Decision Tree Classification Report:

	precision	recall	f1-score	support
0	0.82	0.86	0.84	405
1	0.92	0.90	0.91	755
accuracy			0.88	1160
macro avg	0.87	0.88	0.87	1160
weighted avg	0.89	0.88	0.89	1160

Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.87	0.80	0.83	405
1	0.90	0.93	0.92	755

accuracy			0.89	1160
macro avg	0.88	0.87	0.88	1160
weighted avg	0.89	0.89	0.89	1160

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.84	0.87	0.85	405
1	0.93	0.91	0.92	755
accuracy			0.90	1160
macro avg	0.88	0.89	0.89	1160
weighted avg	0.90	0.90	0.90	1160

REFERENCES

1. "Artificial intelligence modern Application"-Stuart Russel, Peter Norvig
2. <https://youtube.com/playlist?list=PLa6CNrvKM5QU7AjAS90zCMIwi9RTFNIIW&feature=shared>
3. <https://support.zyte.com/support/solutions/articles/22000201935-deploying-a-project-from-a-github-repository>
4. <https://habr.com/ru/articles/510934/>
5. <https://scikit-learn.org/stable/>
6. <https://habr.com/ru/articles/864890/>
7. <https://www.datacamp.com/tutorial/random-forests-classifier-python>
8. <https://asana.com/ru/resources/decision-tree-analysis>
9. <https://habr.com/ru/companies/skillfactory/articles/675876/>
10. <https://medium.com/@thomas.vidori/how-to-create-a-python-package-and-publish-it-on-github-eebc78b2a12d>
11. <https://my.sdu.edu.kz/>