PROBLEM STATEMENT

As Data Scientists, we must develop a model to predict/classify whether the social ad is purchased or not purchased when age,gender,estimated_salary is given as input

Three stages of Problem Identification:

Stage 1- Domain Selection- Machine learning

As the input (Social Ad Dataset) contains numerical values, we can choose the ***Machine Learning Domain***

Stage 2-Learning – Supervised Learning

As the Requirement is clear (predict purchased or not purchased) we can choose ***Supervised Learning***

Stage 3 – Classification:

Prediction related to purchased or not purchased (0 or 1) so we can choose the ***Classification type***

DATASET INFORMATION

Total No of rows: 400

Total No of columns:6

Column Name: User ID, Gender, Age, EstimatedSalary

Input Variables: Gender, Age, EstimatedSalary

Output Variable: Purchased

```
Dataset=pd.read_csv("Social_Network_Ads.csv")
```

```
Dataset
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

PREPROCESSING METHOD:

Here we can drop the user id column which is not significant in classifying the output

```
Dataset=Dataset.drop("User ID",axis=1)
```

```
Dataset
```

	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0
...	...	...	...	...
395	Female	46	41000	1
396	Male	51	23000	1
397	Female	50	20000	1
398	Male	36	33000	0
399	Female	49	36000	1

400 rows × 4 columns

Since the Dataset has categorical columns Gender, it should be converted to numbers

Here the categorical data is nominal, so one hot encoding method is used to convert the categorical data (string) to numerical data (numbers 1 or 0)

```
[11]: Dataset=pd.get_dummies(Dataset,dtype=int,drop_first=True)
```

```
[13]: Dataset
```

[13]:

	Age	EstimatedSalary	Purchased	Gender_Male
0	19	19000	0	1
1	35	20000	0	1
2	26	43000	0	0
3	27	57000	0	0
4	19	76000	0	1
...	...	...	...	...
395	46	41000	1	0
396	51	23000	1	1
397	50	20000	1	0
398	36	33000	0	1
399	49	36000	1	0

400 rows × 4 columns

1. **DECISION TREE :**

S.No	CRITERION	MAX FEATURES	SPLITTER	Random _state	Max_ depth	Recall (macro avg)	Precision (macro avg)	F1-score (mac avg)	Accuracy
1	Gini	Sqrt	best	0	4	0.89	0.88	0.88	0.89
2	Gini	Sqrt	random	None	6	0.81	0.83	0.82	0.84
3	Gini	log2	best	0	4	0.89	0.88	0.88	0.89
4	Gini	log2	random	None	6	0.88	0.87	0.84	0.88
5	Gini	None	best	0	4	0.88	0.88	0.88	0.89
6	Gini	None	random	None	6	0.88	0.91	0.89	0.90
7	Entropy	Sqrt	best	0	4	0.91	0.89	0.90	0.90
8	Entropy	Sqrt	random	None	6	0.80	0.82	0.81	0.83
9	Entropy	log2	best	0	4	0.91	0.89	0.90	0.90
10	Entropy	log2	random	None	6	0.85	0.86	0.85	0.87
11	Entropy	None	best	0	4	0.87	0.89	0.88	0.89
12	Entropy	None	random	None	6	0.88	0.88	0.88	0.89
13	log_loss	Sqrt	best	0	4	0.91	0.89	0.90	0.90
14	log_loss	Sqrt	random	None	6	0.73	0.79	0.74	0.78
15	log_loss	log2	best	0	4	0.91	0.89	0.90	0.90
16	log_loss	log2	random	None	6	0.77	0.80	0.78	0.81
17	log_loss	None	best	0	4	0.87	0.89	0.88	0.89
18	log_loss	None	random	None	6	0.88	0.89	0.89	0.90

Decision Tree has highest **accuracy, recall, precision and f1-score** for max_feature='sqrt', criterion=entropy, splitter='best', random_state=0 and max_depth=4

2. SUPPORT VECTOR MACHINE:

S.NO	HYPERTUNING PARAMETER				KERNEL TYPE (LINEAR)			
	C	Max_iter	gamma	random_state	Recall (macro avg)	Precision (macro avg)	F1-score (mac avg)	Accuracy
1	1.0	-1	auto	0	0.83	0.86	0.84	0.86
2	1.0	-1	scale	None	0.83	0.86	0.84	0.86
3	100	500	auto	0	0.75	0.81	0.76	0.80
4	100	500	scale	None	0.75	0.81	0.76	0.80
5	500	1000	auto	0	0.74	0.83	0.75	0.80
6	500	1000	scale	None	0.74	0.83	0.75	0.80
7	2000	4000	auto	0	0.74	0.83	0.75	0.80
8	2000	4000	scale	None	0.74	0.83	0.75	0.80
9	4000	8000	auto	0	0.74	0.83	0.75	0.80
10	4000	8000	scale	None	0.74	0.83	0.75	0.80

S.NO	HYPERTUNING PARAMETER				KERNEL TYPE (rbf)			
	C	Max_iter	gamma	random_state	Recall (macro avg)	Precision (macro avg)	F1-score (mac avg)	Accuracy
1	1.0	-1	auto	0	0.57	0.77	0.52	0.68
2	1.0	-1	scale	None	0.72	0.82	0.73	0.78
3	100	500	auto	0	0.59	0.79	0.55	0.69
4	100	500	scale	None	0.74	0.83	0.75	0.80
5	500	1000	auto	0	0.59	0.79	0.55	0.69
6	500	1000	scale	None	0.75	0.80	0.77	0.80
7	2000	4000	auto	0	0.59	0.79	0.55	0.69
8	2000	4000	scale	None	0.74	0.83	0.75	0.80
9	4000	8000	auto	0	0.59	0.79	0.55	0.69
10	4000	8000	scale	None	0.74	0.83	0.75	0.80

S.NO	HYPERTUNING PARAMETER				KERNEL TYPE (poly)			
	C	Max_iter	gamma	random_state	Recall (macro avg)	Precision (macro avg)	F1-score (mac avg)	Accuracy
1	1.0	-1	auto	0	0.85	0.89	0.87	0.88
2	1.0	-1	scale	None	0.66	0.78	0.66	0.74
3	100	500	auto	0	0.81	0.79	0.79	0.80
4	100	500	scale	None	0.73	0.83	0.74	0.79
5	500	1000	auto	0	0.58	0.84	0.54	0.69
6	500	1000	scale	None	0.61	0.62	0.55	0.55
7	2000	4000	auto	0	0.58	0.84	0.54	0.69
8	2000	4000	scale	None	0.63	0.63	0.60	0.60
9	4000	8000	auto	0	0.37	0.15	0.21	0.27
10	4000	8000	scale	None	0.72	0.82	0.73	0.78

S.NO	HYPERTUNING PARAMETER				KERNEL TYPE (sigmoid)			
	C	Max_iter	gamma	random_state	Recall (macro avg)	Precision (macro avg)	F1-score (mac avg)	Accuracy
1	1.0	-1	auto	0	0.50	0.32	0.39	0.63
2	1.0	-1	scale	None	0.44	0.44	0.44	0.49
3	100	500	auto	0	0.50	0.32	0.39	0.63
4	100	500	scale	None	0.65	0.64	0.64	0.66
5	500	1000	auto	0	0.50	0.32	0.39	0.63
6	500	1000	scale	None	0.65	0.64	0.64	0.66
7	2000	4000	auto	0	0.50	0.32	0.39	0.63
8	2000	4000	scale	None	0.65	0.64	0.64	0.66
9	4000	8000	auto	0	0.50	0.32	0.39	0.63
10	4000	8000	scale	None	0.65	0.64	0.64	0.66

Support Vector machine has highest **accuracy, recall, precision and f1-score** for c=1.0,max_iter=-1, Gamma='auto', random_state=0 and kernel='poly'

3. RANDOM FOREST:

S.No	CRITERION	max_ features	n_ estimators	Max_ depth	Random _state	Recall (mac avg)	Precision (macro avg)	F1-score (macro avg)	Accuracy
1	gini	None	100	4	0	0.93	0.92	0.92	0.93
2	gini	None	50	5	None	0.91	0.91	0.91	0.92
3	gini	Sqrt	100	4	0	0.92	0.91	0.91	0.92
4	gini	Sqrt	50	5	None	0.92	0.91	0.91	0.92
5	gini	log2	100	4	0	0.92	0.91	0.91	0.92
6	gini	log2	50	5	None	0.92	0.91	0.91	0.92
7	entropy	None	100	4	0	0.93	0.92	0.92	0.93
8	entropy	None	50	5	None	0.88	0.89	0.89	0.90
9	entropy	Sqrt	100	4	0	0.92	0.91	0.91	0.92
10	entropy	Sqrt	50	5	None	0.91	0.90	0.90	0.91
11	entropy	log2	100	4	0	0.92	0.91	0.91	0.92
12	entropy	log2	50	5	None	0.92	0.91	0.91	0.92
13	log_loss	None	100	4	0	0.93	0.92	0.92	0.93
14	log_loss	None	50	5	None	0.88	0.89	0.89	0.90
15	log_loss	Sqrt	100	4	0	0.92	0.91	0.91	0.92
16	log_loss	Sqrt	50	5	None	0.91	0.90	0.90	0.91
17	log_loss	log2	100	4	0	0.92	0.91	0.91	0.92
18	log_loss	log2	50	5	None	0.91	0.90	0.90	0.91

Random Forest Tree has highest **accuracy, recall, precision and f1-score** for max_feature=None, criterion='gini', n_estimators=100, random_state=0 and max_depth=4

3. LOGISTIC REGRESSION :

S.NO	solver	penalty	max_iter	C	Recall (mac avg)	Precision (macro avg)	F1-score (macro	Accuracy

							avg)	
1	lbfgs	l2	100	1.0	0.87	0.88	0.88	0.89
2	lbfgs	None	100	1.0	0.87	0.88	0.88	0.89
3	liblinear	l1	1000	10	0.86	0.88	0.87	0.88
4	liblinear	l2	1000	10	0.82	0.50	0.39	0.63
5	newton-cg	l2	2000	100	0.87	0.88	0.88	0.89
6	newton-cg	None	2000	100	0.87	0.88	0.88	0.89
7	newton-cholesky	l2	3000	1000	0.87	0.88	0.88	0.89
8	newton-cholesky	None	3000	1000	0.87	0.88	0.88	0.89
9	sag	l2	4000	2000	0.82	0.50	0.39	0.63
10	sag	None	4000	2000	0.82	0.50	0.39	0.63
11	saga	l2	5000	3000	0.82	0.50	0.39	0.63
12	saga	None	5000	3000	0.82	0.50	0.39	0.63
13	saga	l1	5000	3000	0.82	0.50	0.39	0.63
14	saga	elasticnet l1_ratio=0.5	5000	3000	0.82	0.50	0.39	0.63

Logistic Regression has highest **accuracy, recall, precision and f1-score** for

solver='lbfgs',penalty=l2,max_jter=100,c=1.0

solver='lbfgs',penalty=None,max_jter=100,c=1.

4. KNEIGHBORS CLASSIFIER:

S.NO	n_ neighbors	weight	algorithm	P	metric	Recall (mac avg)	Precision (macro avg)	F1-score (macro avg)	Accuracy
1	7	uniform	auto	2	minkowski	0.80	0.84	0.81	0.84
2	5	distance	auto	1	minkowski	0.72	0.71	0.71	0.72
3	7	uniform	ball_tree	2	minkowski	0.80	0.84	0.81	0.84
4	5	distance	ball_tree	1	minkowski	0.72	0.71	0.71	0.72
5	7	uniform	kd_tree	2	minkowski	0.80	0.84	0.81	0.84
6	5	distance	kd_tree	1	minkowski	0.72	0.71	0.71	0.72
7	7	uniform	brute	2	minkowski	0.80	0.84	0.81	0.84
8	5	distance	brute	1	minkowski	0.72	0.71	0.72	0.73

MODEL SELECTION:

Out of all the above models***, Random Forest Tree*** has highest **accuracy, recall, precision and f1-score** so it is selected as the best model