University Institute of Engineering

Department of Computer Science & Engineering

Experiment -1.4

: **Build a classification mode by using different machine learning algorithms.**

Student Name: **UID:**
Branch: Computer Science & Engineering **Section/Group:**
Semester: 1st Semester **Date of Performance:**
Subject Name: Disruptive Technologies-1 Subject
Code: 21ECP-102

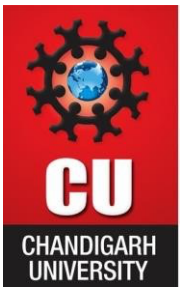
1. **Aim of the practical:** Build a classification mode by using different machine learning algorithms

2. **Tool Used:** Google colab

3. **Basic Concept/ Command Description:**

Python is a powerful general-purpose programming language. Python has simple easyto-use syntax In the experiment performed, the basic concepts and command discussed are as follows:

- Finalize Model: How to finalize the best model at the end of the experiment
- Predict Model: How to make prediction on new / unseen data



University Institute of Engineering

Department of Computer Science & Engineering

4. Code: Install pycaret

```
!pip install pycaret &> /dev/null

print ("Pycaret installed sucessfully!!")
```

Output:

```
Pycaret installed sucessfully!!
```

Code: Loading Dataset - Loading dataset from pycaret

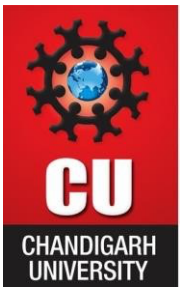
```
from pycaret.datasets import get_data

# No output
```

Code: Get the list of datasets available in pycaret (55)

```
# Internet connection is required dataSets
= get_data('index')
```

Output:

	Dataset	Data Types	Default Task	Target Variable 1	Target Variable 2	# Instances	# Attributes	Missing Values
0	anomaly	Multivariate	Anomaly Detection	None	None	1000	10	N
1	france	Multivariate	Association Rule Mining	InvoiceNo	Description	8557	8	N
2	germany	Multivariate	Association Rule Mining	InvoiceNo	Description	9495	8	N
3	bank	Multivariate	Classification (Binary)	deposit	None	45211	17	N
4	blood	Multivariate	Classification (Binary)	Class	None	748	5	N
5	cancer	Multivariate	Classification (Binary)	Class	None	683	10	N
6	credit	Multivariate	Classification (Binary)	default	None	24000	24	N
7	diabetes	Multivariate	Classification (Binary)	Class variable	None	768	9	N
8	electrical_grid	Multivariate	Classification (Binary)	stabf	None	10000	14	N
9	employee	Multivariate	Classification (Binary)	left	None	14999	10	N
10	heart	Multivariate	Classification (Binary)	DEATH	None	200	16	N



University Institute of Engineering

Department of Computer Science & Engineering

Code: Get diabetes dataset

```
diabetesDataSet = get_data("diabetes")     # SN is 7
# This is binary classification dataset. The values in "Class variable" have two (bin
ary) values.
```

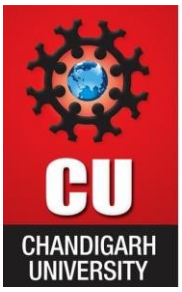
Output:

	Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Code: Parameter setting for all classification models

```
from pycaret.classification import *
s = setup(data=diabetesDataSet, target='Class variable', silent=True)
```

Output:



University Institute of Engineering

Department of Computer Science & Engineering

	Description	Value
0	session_id	3798
1	Target	Class variable
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(768, 9)
5	Missing Values	False
6	Numeric Features	7
7	Categorical Features	1
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None

Code: Run and compare the Model Performance

```
cm = compare_models()
# Explore more parameters
```

Output:

University Institute of Engineering

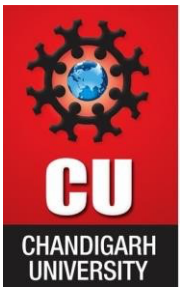
Department of Computer Science & Engineering

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.7636	0.8241	0.6368	0.7047	0.6620	0.4822	0.4893	0.492
gbc	Gradient Boosting Classifier	0.7469	0.8359	0.5953	0.6806	0.6309	0.4404	0.4455	0.116
lda	Linear Discriminant Analysis	0.7467	0.8047	0.5555	0.7081	0.6131	0.4313	0.4435	0.016
ada	Ada Boost Classifier	0.7432	0.8283	0.6105	0.6733	0.6346	0.4383	0.4437	0.099
lr	Logistic Regression	0.7429	0.8094	0.5353	0.7133	0.6009	0.4191	0.4343	0.522
ridge	Ridge Classifier	0.7411	0.0000	0.5355	0.7049	0.5988	0.4156	0.4294	0.012
lightgbm	Light Gradient Boosting Machine	0.7393	0.8173	0.6308	0.6528	0.6382	0.4353	0.4379	0.076
et	Extra Trees Classifier	0.7358	0.8098	0.5611	0.6831	0.6116	0.4145	0.4225	0.457
dt	Decision Tree Classifier	0.7116	0.6887	0.6013	0.6111	0.6048	0.3782	0.3792	0.015
knn	K Neighbors Classifier	0.7002	0.7484	0.5403	0.6089	0.5663	0.3401	0.3451	0.116
nb	Naive Bayes	0.6796	0.7374	0.2624	0.7207	0.3675	0.2118	0.2690	0.014
dummy	Dummy Classifier	0.6313	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.014
qda	Quadratic Discriminant Analysis	0.5737	0.6068	0.4421	0.5220	0.3511	0.0927	0.1384	0.017
svm	SVM - Linear Kernel	0.5528	0.0000	0.6082	0.4104	0.4182	0.1177	0.1400	0.015

Code: Three line of code for model comparison for "Heart Disease" dataset

```
from pycaret.datasets import get_data from pycaret.classification import
*
 heartDiseaseDataSet =
get_data("heart_disease")
s = setup(data = heartDiseaseDataSet, target='Disease', silent=True) cm
= compare_models()
```

Output:



University Institute of Engineering

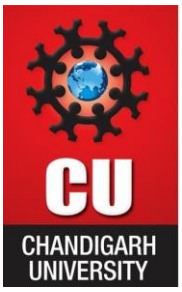
Department of Computer Science & Engineering

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.8515	0.9010	0.7875	0.8429	0.8055	0.6883	0.6980	0.462
lr	Logistic Regression	0.8406	0.8975	0.7464	0.8570	0.7890	0.6642	0.6764	0.145
rf	Random Forest Classifier	0.8301	0.9000	0.7714	0.8183	0.7852	0.6466	0.6575	0.465
lda	Linear Discriminant Analysis	0.8251	0.8997	0.7607	0.8207	0.7803	0.6370	0.6478	0.017
ridge	Ridge Classifier	0.8196	0.0000	0.7339	0.8267	0.7671	0.6225	0.6356	0.013
lightgbm	Light Gradient Boosting Machine	0.8041	0.8792	0.7464	0.7890	0.7580	0.5946	0.6067	0.028
gbc	Gradient Boosting Classifier	0.7825	0.8585	0.7589	0.7362	0.7353	0.5526	0.5671	0.080
nb	Naive Bayes	0.7713	0.8796	0.5036	0.9000	0.6305	0.4938	0.5417	0.015
ada	Ada Boost Classifier	0.7713	0.8669	0.6982	0.7485	0.7020	0.5242	0.5420	0.092
dt	Decision Tree Classifier	0.7301	0.7213	0.6607	0.6811	0.6657	0.4414	0.4450	0.016
knn	K Neighbors Classifier	0.6813	0.6872	0.5571	0.6618	0.5900	0.3370	0.3471	0.116
svm	SVM - Linear Kernel	0.6009	0.0000	0.3250	0.3699	0.2820	0.1247	0.1445	0.014
qda	Quadratic Discriminant Analysis	0.5798	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.020
dummy	Dummy Classifier	0.5798	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.013

Code: Model performance using data "Normalization"

```
s = setup(data=diabetesDataSet, target='Class variable', normalize = True, normalize_
method = 'zscore', silent=True) cm = compare_models()

#normalize_method = {zscore, minmax, maxabs, robust}
```

Output:



University Institute of Engineering

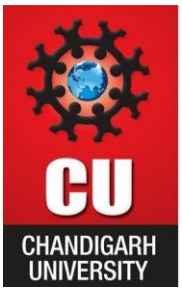
Department of Computer Science & Engineering

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7691	0.8286	0.5640	0.7101	0.6203	0.4600	0.4712	0.025
rf	Random Forest Classifier	0.7654	0.8273	0.5743	0.6871	0.6212	0.4550	0.4613	0.502
ridge	Ridge Classifier	0.7597	0.0000	0.5368	0.7037	0.5999	0.4352	0.4488	0.013
et	Extra Trees Classifier	0.7597	0.8076	0.5523	0.6914	0.6088	0.4401	0.4487	0.461
gbc	Gradient Boosting Classifier	0.7580	0.8194	0.5909	0.6614	0.6189	0.4444	0.4494	0.120
ada	Ada Boost Classifier	0.7578	0.8011	0.5909	0.6755	0.6231	0.4475	0.4543	0.103
lda	Linear Discriminant Analysis	0.7560	0.8210	0.5421	0.6920	0.5992	0.4298	0.4418	0.016
lightgbm	Light Gradient Boosting Machine	0.7524	0.8014	0.6175	0.6423	0.6264	0.4422	0.4448	0.049
knn	K Neighbors Classifier	0.7429	0.7801	0.5205	0.6644	0.5789	0.3991	0.4081	0.118
svm	SVM - Linear Kernel	0.7263	0.0000	0.5310	0.6213	0.5605	0.3676	0.3769	0.017
dt	Decision Tree Classifier	0.7095	0.6855	0.6073	0.5805	0.5872	0.3651	0.3711	0.017
nb	Naive Bayes	0.6815	0.7427	0.2418	0.6020	0.3385	0.1805	0.2137	0.015
dummy	Dummy Classifier	0.6537	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.012
qda	Quadratic Discriminant Analysis	0.5312	0.5660	0.5310	0.3816	0.3742	0.0555	0.0532	0.014

Code: Model Performance using “Feature Selection”

```
s = setup(data=diabetesDataSet, target='Class variable', feature_selection = True, fe
ature_selection_threshold = 0.9, silent=True) cm = compare_models()
```

Output:



University Institute of Engineering

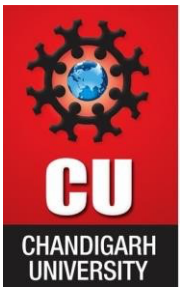
Department of Computer Science & Engineering

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ada	Ada Boost Classifier	0.7580	0.7959	0.6178	0.6789	0.6410	0.4603	0.4662	0.103
lr	Logistic Regression	0.7449	0.8048	0.5313	0.6876	0.5907	0.4111	0.4242	0.242
gbc	Gradient Boosting Classifier	0.7430	0.8116	0.5898	0.6517	0.6138	0.4231	0.4283	0.121
rf	Random Forest Classifier	0.7412	0.8070	0.5310	0.6783	0.5880	0.4043	0.4160	0.508
ridge	Ridge Classifier	0.7393	0.0000	0.5155	0.6899	0.5784	0.3967	0.4132	0.015
lda	Linear Discriminant Analysis	0.7356	0.7962	0.5102	0.6842	0.5728	0.3884	0.4050	0.016
et	Extra Trees Classifier	0.7207	0.7758	0.5000	0.6447	0.5577	0.3582	0.3686	0.462
knn	K Neighbors Classifier	0.7135	0.7400	0.5380	0.6048	0.5678	0.3550	0.3576	0.118
lightgbm	Light Gradient Boosting Machine	0.7134	0.7868	0.5465	0.6022	0.5705	0.3566	0.3591	0.046
dt	Decision Tree Classifier	0.6927	0.6697	0.5944	0.5618	0.5735	0.3346	0.3383	0.016
nb	Naive Bayes	0.6667	0.7316	0.2664	0.6243	0.3600	0.1730	0.2140	0.015
qda	Quadratic Discriminant Analysis	0.6499	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.018
dummy	Dummy Classifier	0.6499	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.014
svm	SVM - Linear Kernel	0.6012	0.0000	0.1629	0.2902	0.1278	0.0058	0.0057	0.014

Code: Model Performance using “Transformation”

```
s = setup(data=diabetesDataSet, target='Class variable', transformation = True, trans
formation_method = 'yeo-johnson', silent=True) cm = compare_models()
```

Output:



University Institute of Engineering

Department of Computer Science & Engineering

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7766	0.8169	0.5462	0.7308	0.6208	0.4677	0.4804	0.024
ridge	Ridge Classifier	0.7636	0.0000	0.5246	0.7030	0.5980	0.4360	0.4470	0.013
rf	Random Forest Classifier	0.7636	0.8112	0.5251	0.6980	0.5976	0.4356	0.4453	0.507
lda	Linear Discriminant Analysis	0.7617	0.8145	0.5357	0.6910	0.6016	0.4359	0.4442	0.016
lightgbm	Light Gradient Boosting Machine	0.7467	0.7995	0.5529	0.6522	0.5938	0.4124	0.4185	0.046
ada	Ada Boost Classifier	0.7466	0.7983	0.5582	0.6485	0.5983	0.4149	0.4187	0.102
gbc	Gradient Boosting Classifier	0.7430	0.8033	0.5360	0.6472	0.5836	0.4007	0.4062	0.118
et	Extra Trees Classifier	0.7411	0.7824	0.5032	0.6481	0.5652	0.3857	0.3925	0.462
knn	K Neighbors Classifier	0.7261	0.7659	0.4523	0.6340	0.5265	0.3413	0.3519	0.118
dt	Decision Tree Classifier	0.7095	0.6685	0.5421	0.5820	0.5564	0.3421	0.3460	0.016
svm	SVM - Linear Kernel	0.7094	0.0000	0.5406	0.6065	0.5542	0.3432	0.3549	0.015
nb	Naive Bayes	0.6965	0.7492	0.3757	0.5628	0.4451	0.2526	0.2630	0.015
dummy	Dummy Classifier	0.6629	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.013
qda	Quadratic Discriminant Analysis	0.4284	0.4627	0.6389	0.2899	0.3592	-0.0323	-0.0775	0.014

Three lines of code for model comparison for “cancer” dataset

```
from pycaret.datasets import get_data
from pycaret.classification import *

cancerDataSet = get_data("cancer")
s = setup(data = cancerDataSet, target='Class', silent=True)
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.9603	0.0000	0.9581	0.9343	0.9442	0.9134	0.9157	0.015
rf	Random Forest Classifier	0.9582	0.9901	0.9518	0.9329	0.9407	0.9085	0.9104	0.464
lr	Logistic Regression	0.9561	0.9881	0.9397	0.9377	0.9375	0.9036	0.9050	0.026
svm	SVM - Linear Kernel	0.9539	0.0000	0.9338	0.9399	0.9347	0.8992	0.9016	0.019
lda	Linear Discriminant Analysis	0.9519	0.9769	0.9338	0.9329	0.9313	0.8943	0.8967	0.025
et	Extra Trees Classifier	0.9519	0.9907	0.9390	0.9257	0.9309	0.8940	0.8956	0.461
nb	Naive Bayes	0.9518	0.9717	0.9574	0.9170	0.9336	0.8959	0.9002	0.017
knn	K Neighbors Classifier	0.9457	0.9790	0.8915	0.9517	0.9196	0.8786	0.8807	0.120
lightgbm	Light Gradient Boosting Machine	0.9456	0.9867	0.9393	0.9112	0.9236	0.8814	0.8833	0.045
ada	Ada Boost Classifier	0.9435	0.9807	0.9044	0.9364	0.9176	0.8748	0.8778	0.107
gbc	Gradient Boosting Classifier	0.9435	0.9863	0.9092	0.9304	0.9174	0.8745	0.8771	0.117
dt	Decision Tree Classifier	0.9079	0.8929	0.8434	0.8890	0.8629	0.7938	0.7970	0.017
qda	Quadratic Discriminant Analysis	0.8683	0.8972	0.9941	0.7501	0.8492	0.7405	0.7698	0.022

Three lines of code for model comparison for “Heart disease ” dataset

```
from pycaret.datasets import get_data
from pycaret.classification import *

heartDiseaseDataSet = get_data("heart_disease")
s = setup(data = heartDiseaseDataSet, target='Disease', silent=True)
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.8775	0.9278	0.8089	0.8909	0.8432	0.7437	0.7512	0.462
lr	Logistic Regression	0.8673	0.9122	0.7821	0.8983	0.8295	0.7219	0.7349	0.208
lda	Linear Discriminant Analysis	0.8673	0.9143	0.7696	0.9000	0.8254	0.7201	0.7297	0.016
ridge	Ridge Classifier	0.8617	0.0000	0.7554	0.8967	0.8151	0.7068	0.7176	0.015
rf	Random Forest Classifier	0.8453	0.9160	0.7554	0.8610	0.7957	0.6736	0.6854	0.461
lightgbm	Light Gradient Boosting Machine	0.8243	0.9088	0.7429	0.8173	0.7759	0.6321	0.6363	0.030
nb	Naive Bayes	0.8143	0.8948	0.8732	0.7414	0.7964	0.6287	0.6452	0.015
ada	Ada Boost Classifier	0.8035	0.8390	0.7161	0.7938	0.7474	0.5875	0.5939	0.095
gbc	Gradient Boosting Classifier	0.7927	0.8795	0.6911	0.7964	0.7325	0.5647	0.5745	0.084
dt	Decision Tree Classifier	0.7187	0.7095	0.6554	0.6694	0.6495	0.4166	0.4279	0.015
knn	K Neighbors Classifier	0.6760	0.6971	0.5625	0.6220	0.5793	0.3203	0.3275	0.113
qda	Quadratic Discriminant Analysis	0.6602	0.5804	0.4607	0.4682	0.4274	0.2555	0.2910	0.016
svm	SVM - Linear Kernel	0.6325	0.0000	0.7375	0.5926	0.5992	0.2777	0.3328	0.016

Model Transformation using data “Normalization ”

University Institute of Engineering

Department of Computer Science & Engineering

```
s = setup(data=diabetesDataSet, target='Class variable', normalize = True, normalize_method = 'zscore', silent=True)
cm = compare_models()

#normalize_method = {zscore, minmax, maxabs, robust}
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.7730	0.8185	0.5702	0.7320	0.6330	0.4746	0.4868	0.462
lda	Linear Discriminant Analysis	0.7713	0.8175	0.5816	0.7133	0.6338	0.4727	0.4813	0.017
rf	Random Forest Classifier	0.7711	0.8347	0.5751	0.7268	0.6326	0.4717	0.4850	0.504
ridge	Ridge Classifier	0.7694	0.0000	0.5708	0.7151	0.6283	0.4667	0.4761	0.013
lr	Logistic Regression	0.7676	0.8268	0.5760	0.7115	0.6273	0.4640	0.4744	0.026
ada	Ada Boost Classifier	0.7490	0.8006	0.6225	0.6533	0.6328	0.4430	0.4473	0.105
knn	K Neighbors Classifier	0.7487	0.7572	0.5591	0.6688	0.6027	0.4238	0.4303	0.116
lightgbm	Light Gradient Boosting Machine	0.7432	0.8079	0.6173	0.6402	0.6229	0.4297	0.4341	0.050
gbc	Gradient Boosting Classifier	0.7431	0.8198	0.5737	0.6628	0.6058	0.4185	0.4274	0.122
dt	Decision Tree Classifier	0.6855	0.6589	0.5699	0.5520	0.5569	0.3146	0.3173	0.017
svm	SVM - Linear Kernel	0.6835	0.0000	0.5234	0.5490	0.5270	0.2927	0.2987	0.016
nb	Naive Bayes	0.6444	0.7416	0.0579	0.4833	0.1003	0.0228	0.0560	0.015
qda	Quadratic Discriminant Analysis	0.5361	0.5767	0.6570	0.5088	0.4703	0.1300	0.1539	0.016

Model Transformation using data “Feature selection ”

```
[ ]  s = setup(data=diabetesDataSet, target='Class variable', feature_selection = True, feature_selection_threshold = 0.9, silent=True)
     cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7617	0.8230	0.5535	0.6994	0.6140	0.4455	0.4546	0.230
ridge	Ridge Classifier	0.7579	0.0000	0.5319	0.6972	0.5993	0.4311	0.4418	0.014
lda	Linear Discriminant Analysis	0.7579	0.8188	0.5535	0.6903	0.6085	0.4374	0.4468	0.018
ada	Ada Boost Classifier	0.7412	0.7797	0.5626	0.6433	0.5934	0.4061	0.4127	0.106
rf	Random Forest Classifier	0.7374	0.7895	0.4982	0.6584	0.5637	0.3816	0.3914	0.504
lightgbm	Light Gradient Boosting Machine	0.7300	0.7875	0.5632	0.6233	0.5851	0.3870	0.3929	0.047
et	Extra Trees Classifier	0.7206	0.7700	0.4599	0.6299	0.5272	0.3365	0.3475	0.461
knn	K Neighbors Classifier	0.7189	0.7434	0.5529	0.6028	0.5702	0.3632	0.3685	0.118
gbc	Gradient Boosting Classifier	0.7004	0.7692	0.4819	0.5707	0.5201	0.3054	0.3088	0.122
nb	Naive Bayes	0.6815	0.7397	0.2442	0.5638	0.3318	0.1723	0.1980	0.016
dt	Decision Tree Classifier	0.6668	0.6292	0.5137	0.5350	0.5160	0.2637	0.2682	0.016
svm	SVM - Linear Kernel	0.6334	0.0000	0.3234	0.4391	0.2951	0.1223	0.1679	0.016
qda	Quadratic Discriminant Analysis	0.4680	0.5552	0.8342	0.4061	0.5199	0.1031	0.1266	0.016

Model Transformation using data “outlier removal ”

```
[ ]  s = setup(data=diabetesDataSet, target='Class variable', remove_outliers = True, outliers_threshold = 0.05, silent=True)
     cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.7882	0.8265	0.5478	0.7458	0.6237	0.4826	0.4986	0.204
ridge	Ridge Classifier	0.7843	0.0000	0.5357	0.7338	0.6111	0.4694	0.4851	0.013
lda	Linear Discriminant Analysis	0.7804	0.8125	0.5357	0.7233	0.6080	0.4622	0.4766	0.016
rf	Random Forest Classifier	0.7686	0.8019	0.5173	0.6972	0.5910	0.4354	0.4465	0.498
gbc	Gradient Boosting Classifier	0.7608	0.8105	0.5419	0.6629	0.5924	0.4268	0.4335	0.119
et	Extra Trees Classifier	0.7569	0.7824	0.4746	0.6824	0.5571	0.3979	0.4115	0.461
knn	K Neighbors Classifier	0.7451	0.7498	0.5364	0.6227	0.5729	0.3942	0.3983	0.116
ada	Ada Boost Classifier	0.7451	0.7925	0.5357	0.6399	0.5778	0.3974	0.4045	0.105
lightgbm	Light Gradient Boosting Machine	0.7431	0.7921	0.5426	0.6282	0.5768	0.3951	0.4010	0.046
nb	Naive Bayes	0.7000	0.7340	0.4812	0.5535	0.5008	0.2925	0.3021	0.016
dt	Decision Tree Classifier	0.6784	0.6281	0.4853	0.5088	0.4934	0.2594	0.2613	0.017
svm	SVM - Linear Kernel	0.5725	0.0000	0.4743	0.4057	0.3537	0.0888	0.1044	0.018
qda	Quadratic Discriminant Analysis	0.5608	0.5916	0.5081	0.3126	0.3452	0.0871	0.0984	0.016

Model Transformation using data “Transformation” :

2.4 Model Performance using "Transformation"

```
s = setup(data=diabetesDataSet, target='Class variable', transformation = True, transformation_method = 'yeo-johnson', silent=True)
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ridge	Ridge Classifier	0.7636	0.0000	0.5563	0.7200	0.6238	0.4566	0.4668	0.014
lda	Linear Discriminant Analysis	0.7598	0.8122	0.5513	0.7126	0.6181	0.4481	0.4579	0.018
lr	Logistic Regression	0.7579	0.8178	0.5413	0.7146	0.6117	0.4419	0.4533	0.027
svm	SVM - Linear Kernel	0.7449	0.0000	0.6197	0.6580	0.6295	0.4372	0.4440	0.016
rf	Random Forest Classifier	0.7374	0.8000	0.5413	0.6703	0.5931	0.4039	0.4126	0.504
lightgbm	Light Gradient Boosting Machine	0.7317	0.7711	0.6197	0.6288	0.6181	0.4132	0.4177	0.047
gbc	Gradient Boosting Classifier	0.7298	0.8019	0.5621	0.6403	0.5922	0.3936	0.3991	0.117
ada	Ada Boost Classifier	0.7262	0.7872	0.6034	0.6245	0.6068	0.3985	0.4032	0.102
knn	K Neighbors Classifier	0.7242	0.7389	0.4953	0.6563	0.5584	0.3658	0.3767	0.114
et	Extra Trees Classifier	0.7170	0.7697	0.5003	0.6373	0.5548	0.3535	0.3623	0.463
dt	Decision Tree Classifier	0.6816	0.6532	0.5524	0.5505	0.5405	0.3012	0.3077	0.018
nb	Naive Bayes	0.6369	0.7396	0.0468	0.2233	0.0772	0.0140	0.0162	0.016
qda	Quadratic Discriminant Analysis	0.5491	0.5419	0.5139	0.3917	0.4084	0.0766	0.0889	0.018

2.5 Model Performance using "PCA"

Model transformation using “PCA” :

2.5 Model Performance using "PCA"

```
[ ]  s = setup(data=diabetesDataSet, target='Class variable', pca = True, pca_method = 'linear', silent=True)
     cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lda	Linear Discriminant Analysis	0.7451	0.7822	0.4789	0.7037	0.5651	0.3963	0.4128	0.015
lr	Logistic Regression	0.7432	0.7819	0.4789	0.7004	0.5632	0.3928	0.4093	0.020
ridge	Ridge Classifier	0.7414	0.0000	0.4632	0.7009	0.5533	0.3844	0.4021	0.012
gbc	Gradient Boosting Classifier	0.7412	0.7768	0.5526	0.6581	0.5972	0.4102	0.4156	0.105
nb	Naive Bayes	0.7373	0.7765	0.5053	0.6801	0.5772	0.3924	0.4035	0.015
qda	Quadratic Discriminant Analysis	0.7373	0.7815	0.4947	0.6860	0.5714	0.3890	0.4023	0.014
rf	Random Forest Classifier	0.7301	0.7756	0.5368	0.6438	0.5786	0.3847	0.3918	0.517
et	Extra Trees Classifier	0.7189	0.7649	0.5053	0.6257	0.5557	0.3550	0.3608	0.461
ada	Ada Boost Classifier	0.7133	0.7275	0.5105	0.6203	0.5566	0.3484	0.3542	0.102
lightgbm	Light Gradient Boosting Machine	0.7042	0.7501	0.5368	0.6076	0.5629	0.3423	0.3485	0.047
knn	K Neighbors Classifier	0.7022	0.7242	0.5105	0.5890	0.5459	0.3267	0.3289	0.115
dt	Decision Tree Classifier	0.6741	0.6465	0.5526	0.5360	0.5413	0.2894	0.2913	0.016
svm	SVM - Linear Kernel	0.6350	0.0000	0.5158	0.5233	0.5000	0.2213	0.2276	0.014

Model performance using “Outlier Removal” + “Normalization ”:

University Institute of Engineering

Department of Computer Science & Engineering

Model transformation using "Outlier Removal " + "Normalization" +

2.6 Model Performance using "Outlier Removal" + "Normalization"

```
[ ] s = setup(data=diabetesDataSet, target='Class variable', remove_outliers = True, outliers_threshold = 0.05, normalize = True, normalize_method = 'zscore', silent=True)
    cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.7804	0.8402	0.5850	0.7334	0.6461	0.4903	0.5003	0.503
et	Extra Trees Classifier	0.7706	0.8339	0.5725	0.7090	0.6293	0.4666	0.4750	0.461
ada	Ada Boost Classifier	0.7647	0.8035	0.6095	0.6817	0.6399	0.4669	0.4710	0.104
gbc	Gradient Boosting Classifier	0.7627	0.8221	0.5912	0.6828	0.6287	0.4568	0.4627	0.117
lr	Logistic Regression	0.7608	0.8299	0.5624	0.7027	0.6174	0.4474	0.4582	0.022
lightgbm	Light Gradient Boosting Machine	0.7608	0.8085	0.6095	0.6762	0.6352	0.4594	0.4652	0.050
ridge	Ridge Classifier	0.7588	0.0000	0.5680	0.6906	0.6152	0.4437	0.4532	0.014
lda	Linear Discriminant Analysis	0.7588	0.8240	0.5739	0.6879	0.6174	0.4452	0.4543	0.016
knn	K Neighbors Classifier	0.7569	0.7815	0.5794	0.6702	0.6184	0.4426	0.4470	0.117
svm	SVM - Linear Kernel	0.7412	0.0000	0.5905	0.6504	0.6061	0.4165	0.4261	0.017
nb	Naive Bayes	0.7196	0.7678	0.5059	0.6184	0.5538	0.3528	0.3582	0.015
dt	Decision Tree Classifier	0.6941	0.6731	0.6036	0.5525	0.5716	0.3361	0.3413	0.016
qda	Quadratic Discriminant Analysis	0.6098	0.5809	0.3271	0.3601	0.3098	0.0910	0.1028	0.016

"Transformation":

2.7 Model Performance using "Outlier Removal" + "Normalization" + "Transformation"

```
s = setup(data=diabetesDataSet, target='Class variable', remove_outliers = True, outliers_threshold = 0.05, normalize = True, normalize_method = 'zscore', transformation = True, transformation_method = 'yeo-johnson', silent=True)
cm = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ada	Ada Boost Classifier	0.7471	0.7826	0.5471	0.6620	0.5884	0.4093	0.4200	0.104
lr	Logistic Regression	0.7431	0.8059	0.5000	0.6535	0.5585	0.3843	0.3954	0.022
gbc	Gradient Boosting Classifier	0.7431	0.7908	0.5294	0.6412	0.5770	0.3957	0.4012	0.117
lda	Linear Discriminant Analysis	0.7353	0.7998	0.4765	0.6406	0.5397	0.3617	0.3729	0.016
rf	Random Forest Classifier	0.7333	0.7757	0.4588	0.6517	0.5314	0.3541	0.3685	0.511
ridge	Ridge Classifier	0.7314	0.0000	0.4588	0.6377	0.5266	0.3483	0.3609	0.013
et	Extra Trees Classifier	0.7235	0.7460	0.3882	0.6389	0.4782	0.3077	0.3268	0.461
knn	K Neighbors Classifier	0.7176	0.7126	0.4235	0.6116	0.4942	0.3098	0.3223	0.119
nb	Naive Bayes	0.7078	0.7247	0.4588	0.5680	0.5039	0.3031	0.3073	0.015
lightgbm	Light Gradient Boosting Machine	0.7000	0.7566	0.5059	0.5527	0.5216	0.3064	0.3107	0.050
svm	SVM - Linear Kernel	0.6902	0.0000	0.5059	0.5484	0.5136	0.2914	0.2988	0.016
dt	Decision Tree Classifier	0.6667	0.6250	0.5000	0.4913	0.4929	0.2459	0.2473	0.016
qda	Quadratic Discriminant Analysis	0.6588	0.6509	0.3882	0.5124	0.4059	0.1893	0.2018	0.017

University Institute of Engineering

Department of Computer Science & Engineering

6. Additional Creative Inputs (If Any):

Learning outcomes (What I have learnt):

- **Getting Data: How to import data from PyCaret repository**
- **Setting up Environment: How to setup an experiment in PyCaret and get started with building regression models**
- **Create Model: How to create a model, perform cross validation and evaluate regression metrics**
- **Tune Model: How to automatically tune the hyperparameters of a regression model**
- **Plot Model: How to analyze model performance using various plots**
- **Finalize Model: How to finalize the best model at the end of the experiment**
- **Predict Model: How to make prediction on new / unseen data ● Save / Load Model: How to save / load a model for future use**

Evaluation Grid (To be filled by Faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writinglearning objectives/Outcomes.(To besubmitted at the end of the day)		10
2.	Post Lab Quiz Result.		5
3.	Student Engagement in Simulation/Demonstration/Performanc e and Controls/Pre-Lab Questions.		5
	Signature of Faculty (with Date):	**Total Marks Obtained:**	**20**