

BM 336546 - HW2: Type 1 Diabetes (also known as juvenile diabetes)

T1D is a chronic condition resulting from a lack of insulin in the body. The disease typically presents in early childhood or adolescence. Up to 0.33% of the global population suffers from T1D, making it a world-wide and wide-spread issue. There is no cure and the current treatment is to control blood glucose levels through glucose monitoring, insulin injections, diet, and lifestyle modifications to prevent complications. The exact cause of T1D is a mystery; however, there are few possible causes such as genetics, autoimmune dysfunction or environmental factors such as some kind of viruses.

Credit: <https://www.mayoclinic.org/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011>

Assignment

This assignment has no prebuilt notebook. You are required to build and present an appropriate notebook to show your experiments and results. Please provide all answers within the notebook (in a markdown cell), labeled carefully based on the question number. In this assignment, you will do the following: Explore the data provided. Implement linear and non-linear classifiers. Model optimization with k-fold cross validation Evaluate your model performances with appropriate metrics. Present a 2d visualization of multi-featured data. Use feature selection tools. Use the provided HW2 environment and any additional packages you need for this assignment.

Theory Questions

- 1) In most cases model performances is more important than model accuracy. when our dataset is imbalance (A very common case, and also the case for the T1D data set) the accuracy measure can give us misleading result. for example when our dataset contains mainly positive cases. a 'naive' model (which predicts always yes) can get relatively good accuracy result even though it is useless.
- 2) training a model with less feature has some significant benefits: first 1D, 2D and 3D data can be visualized while data with more dimension can't. secondly more feature demands more sample to avoid overfitting. in addition with large data set computing time becomes an important factor. The downside is possibly worse accuracy and performances. more features is more data, some features can have unpredictable correlation with the label, and so can improve the model performances significantly.
- 3) We can assume that the data isn't linearly separable because the biopsies are really similar, and it is difficult to distinguish them from the human eye. Thus, we will use nonlinear SVM to find the correct boundary line.
- 4) SVM tries to create the margin that separates the classes, while LR doesn't. LR returns the probabilities to be in certain class. hyperparameters of SVMs - C is the trade-off between the SVM's goals (increase the distance of decision boundary to classes and Maximize the number of points that are correctly classified in the training set.) if the value of C is low our model may become underfitting and for high value our model may become overfitting.



Coding Assignment

sec 1

first we start by loading the data

```
In [1]: import pandas as pd
import numpy as np
from pathlib import Path
import matplotlib.pyplot as plt

# sec 1 - Load data
file = Path.cwd().joinpath('HW2_data.csv')
T1D_dataset = pd.read_csv(file)
T1D_dataset.head()
```

```
Out[1]:
```

	Age	Gender	Increased Urination	Increased Thirst	Sudden Weight Loss	Weakness	Increased Hunger	Genital Thrush	Visual Blurring	Itching	Irritability
0	45	Male	No	No	No	Yes	No	No	No	Yes	↑
1	42	Male	No	No	No	No	No	No	No	No	↑
2	45	Male	Yes	Yes	No	Yes	No	Yes	No	No	↑
3	59	Female	No	No	No	No	No	No	No	No	↑
4	40	Female	Yes	Yes	Yes	Yes	No	No	Yes	Yes	↑

Secondly we will start with basic analysis and processing of the Data

Given the binary nature of the Data and the low number of nan we decided to drop every row with 'nan' in one of the features

```
In [2]: # sec 1 - Data preprocessing
T1D_dataset_clean = T1D_dataset.dropna()
# print(len(T1D_dataset_clean.index))
# print(len(T1D_dataset.index))
lost_sampels = len(T1D_dataset.index)-len(T1D_dataset_clean.index)
lost_sampels
# number of lost sampel is relatively low
```

```
Out[2]: 42
```

for this section we decided not to do anymore preprocessing. in sec 3 and 4, there is further processing of the data to support the visualzeation and training stages. All this is done after train-test split

sec 2

In this section we split our data into train and test

```
In [3]: from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
# Sec 2

Diagnosis = T1D_dataset_clean.loc[:, 'Diagnosis']
T1D_dataset_clean_no_label = T1D_dataset_clean.drop(['Diagnosis'], axis=1)
T1D_dataset_clean_no_label.head()
X_train, X_test, Y_train, Y_test = train_test_split(T1D_dataset_clean_no_label, np.ravel
                                                    random_state=0, stratify=np.ravel
X_train
```

Out[3]:

	Age	Gender	Increased Urination	Increased Thirst	Sudden Weight Loss	Weakness	Increased Hunger	Genital Thrush	Visual Blurring	Itching	Irrital
406	40	Female	Yes	Yes	Yes	Yes	No	No	Yes	No	
199	68	Male	Yes	Yes	No	No	Yes	No	Yes	Yes	
85	38	Male	Yes	Yes	No	No	Yes	Yes	No	Yes	
312	40	Male	Yes	No	No	No	No	Yes	No	Yes	
166	59	Female	No	No	No	Yes	No	No	Yes	Yes	
...	
284	57	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	No	
270	38	Male	No	No	No	No	No	No	No	No	
190	60	Female	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	
223	38	Male	Yes	No	Yes	Yes	No	No	No	No	
402	65	Female	No	No	No	No	No	Yes	No	No	

418 rows × 17 columns



```
In [4]: from DataExplorer import age_scale

X_train, X_test = age_scale(X_train, X_test)
```

sec 3


a) analysis of the distribution of the features between test and train
 bin_to_num transform our Binary data into 1/0 numeric data with support test/train analysis implementation


```
In [5]: from DataExplorer import bin_to_num
from DataExplorer import test_train_comparison

comparison_data = test_train_comparison(bin_to_num(X_train), bin_to_num(X_test))
comparison_data
```

Out[5]:

	train %	test %
Gender	63.0	62.0
Increased Urination	50.0	49.0
Increased Thirst	45.0	41.0
Sudden Weight Loss	41.0	44.0
Weakness	58.0	59.0
Increased Hunger	46.0	45.0
Genital Thrush	23.0	20.0
Visual Blurring	45.0	43.0
Itching	49.0	46.0
Irritability	25.0	21.0
Delayed Healing	47.0	40.0
Partial Paresis	43.0	44.0
Muscle Stiffness	37.0	40.0
Hair Loss	35.0	34.0
Obesity	16.0	21.0
Family History	49.0	57.0

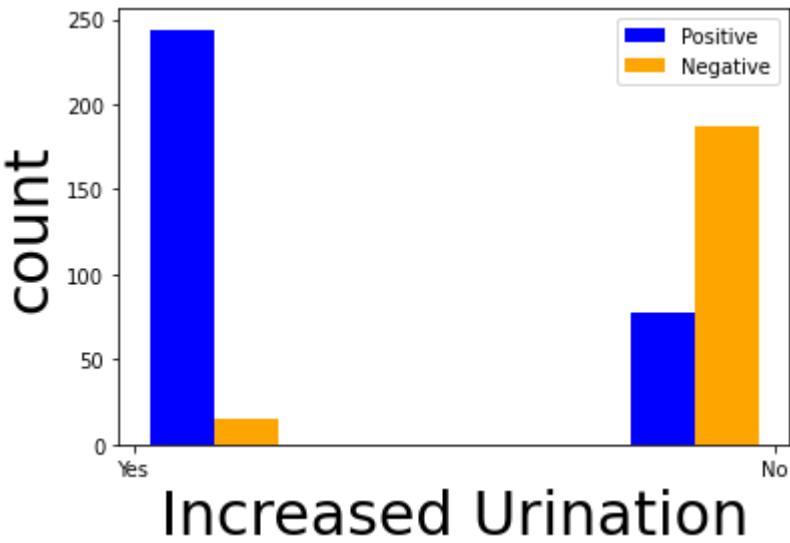
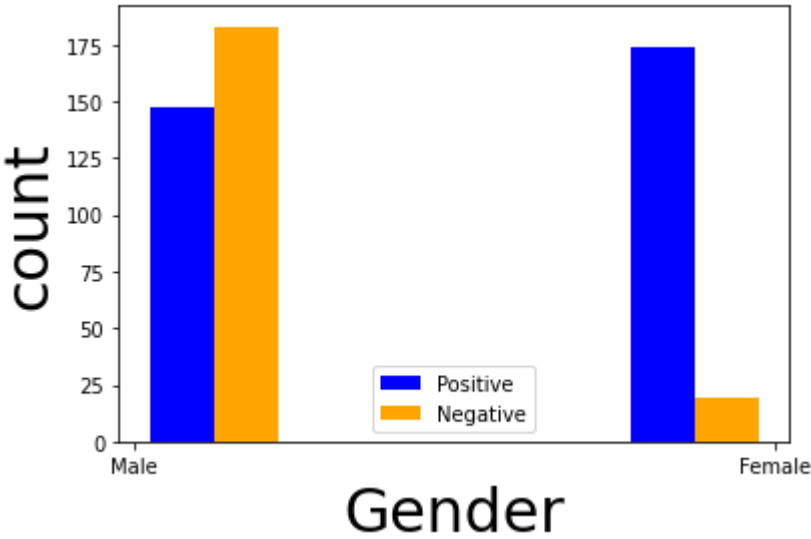
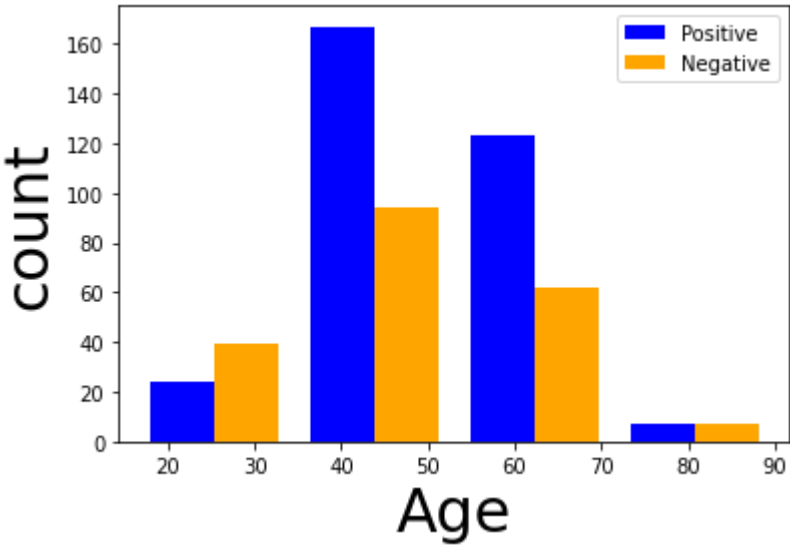
 a.1 An imbalance of features between train and test cause an imbalanced weights and under-fit model.

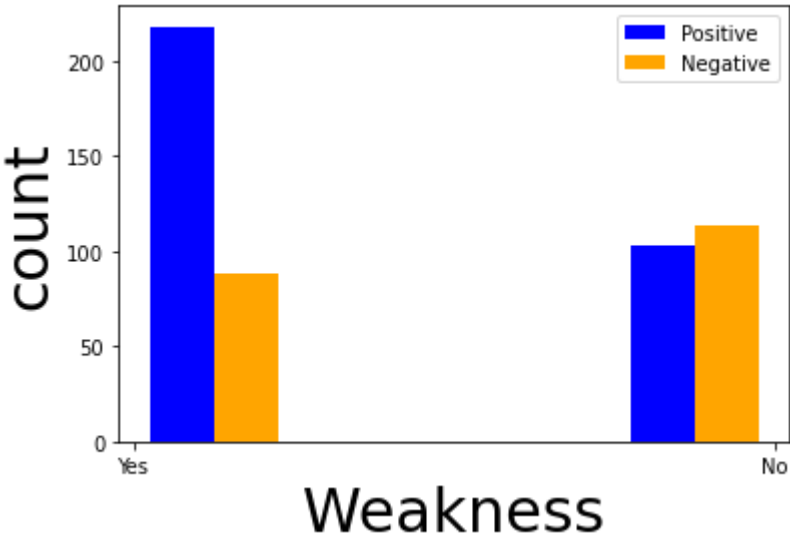
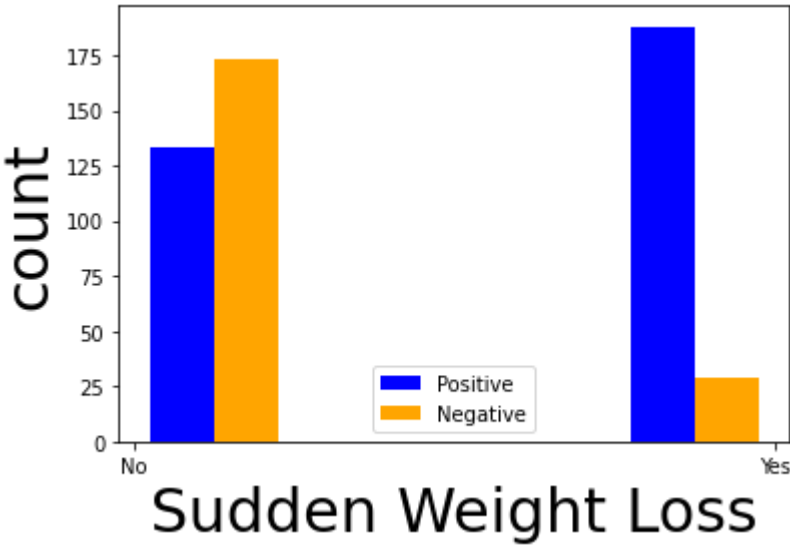
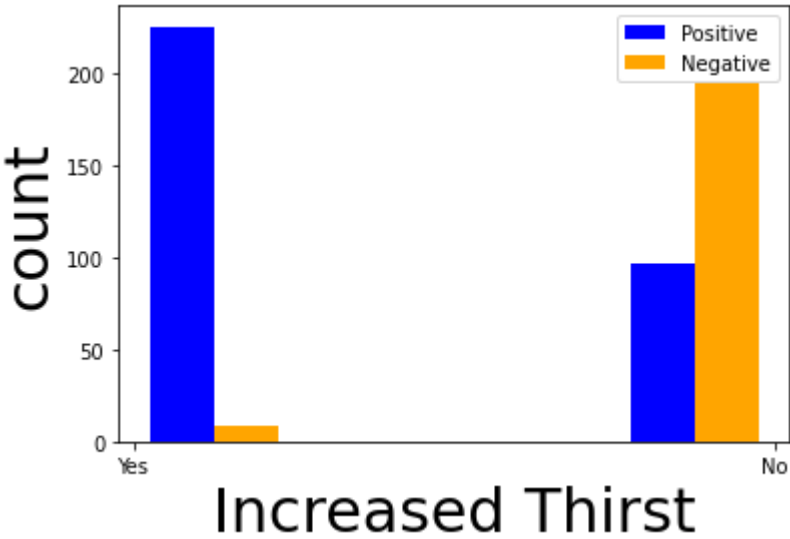
 a.2 In the an imbalanced dataset, one optionis to use different metric for performance measurement or re-sampling technique.As a result of that we will get more suitable devision of the features.

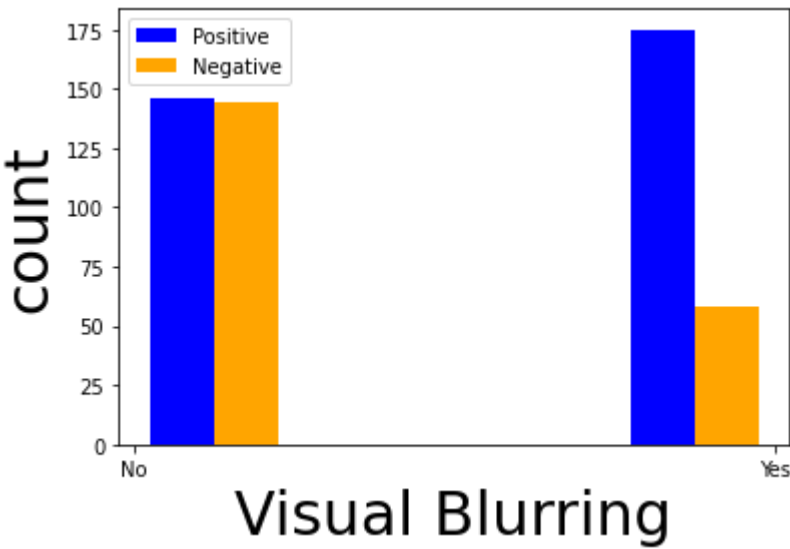
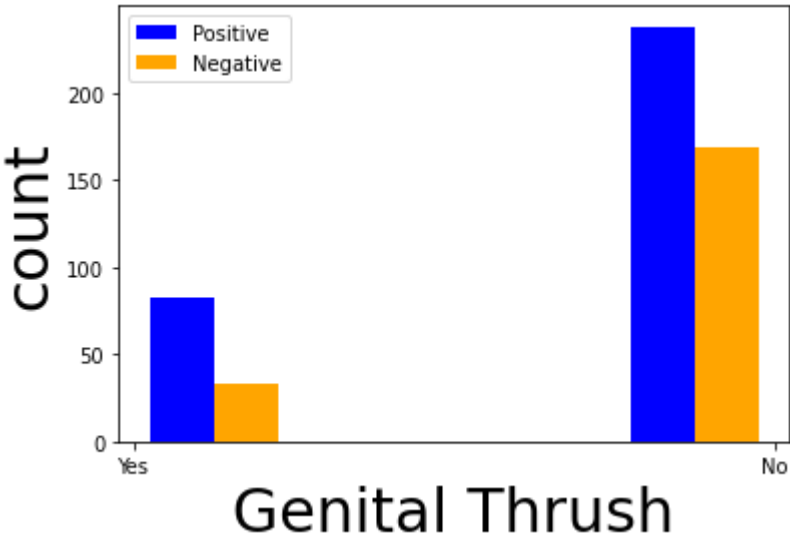
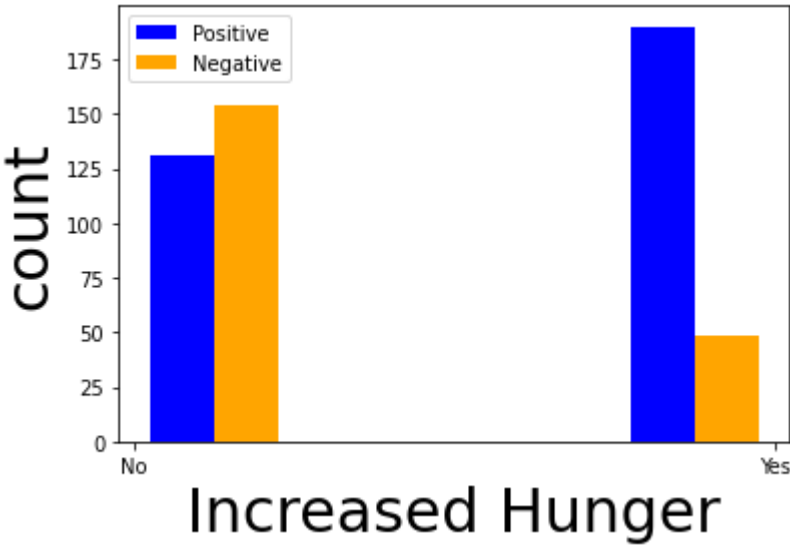
b. relationship between feature and label:

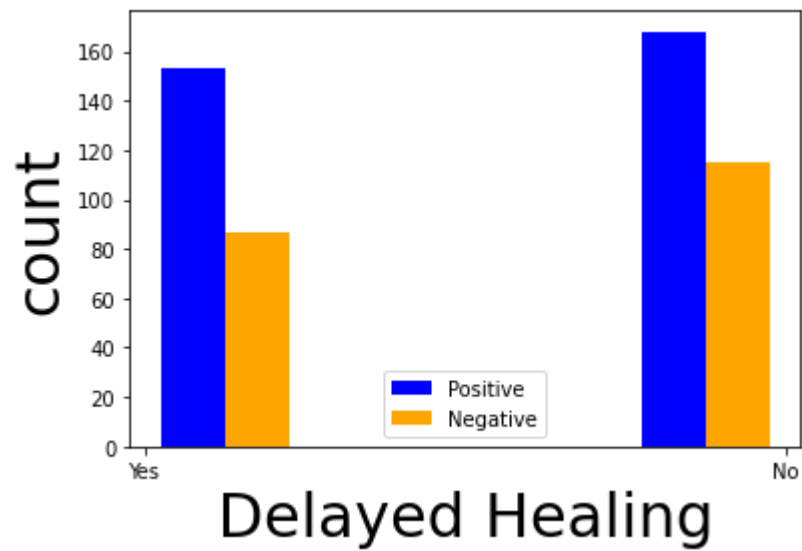
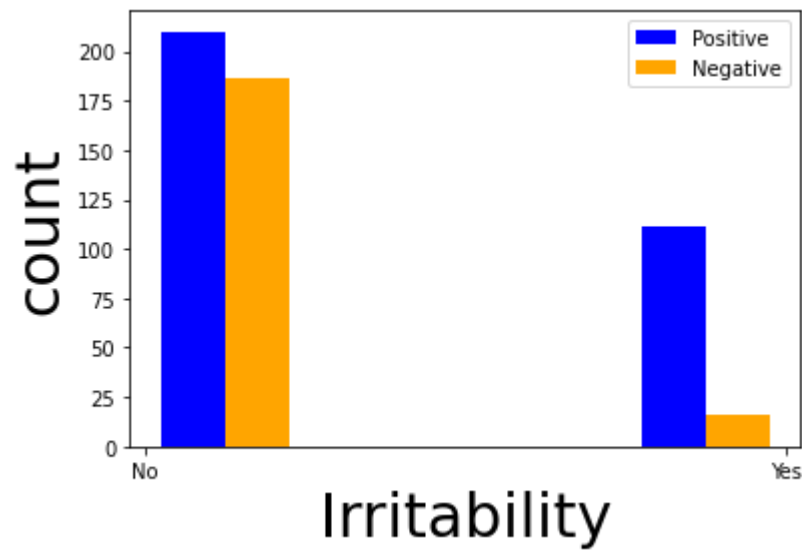
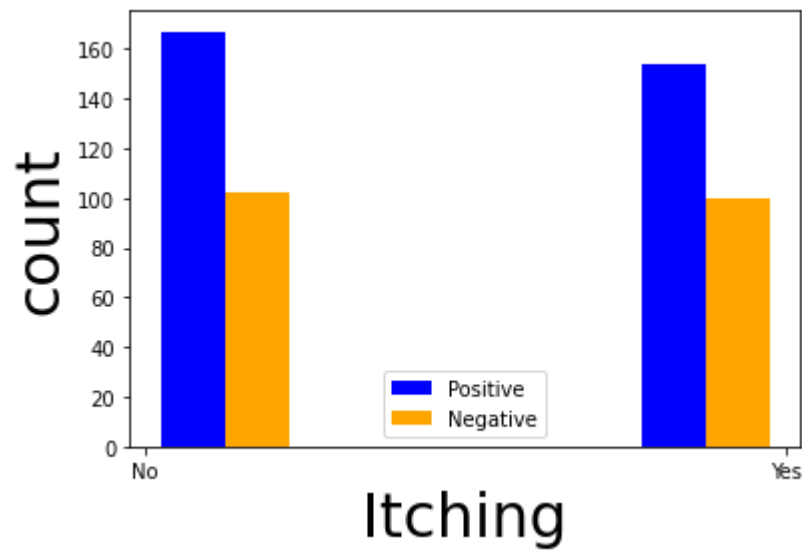
In [6]:

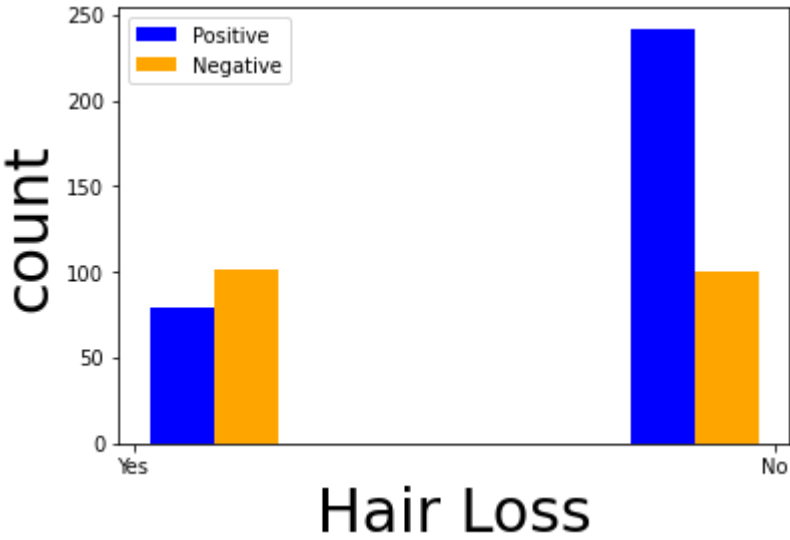
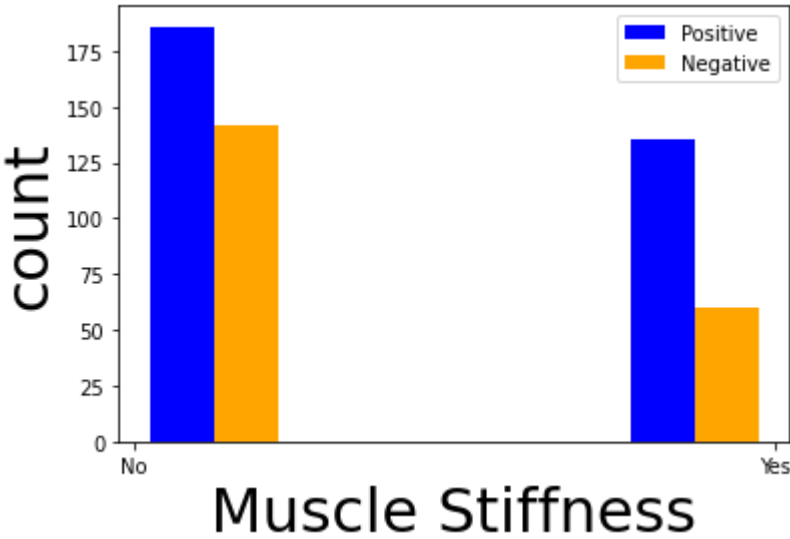
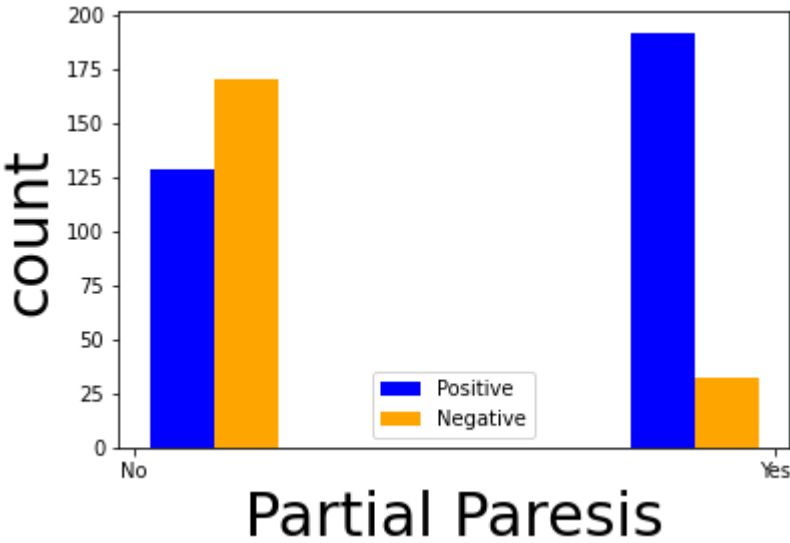
```
from DataExplorer import data_explorer
data_explorer(T1D_dataset_clean)
```

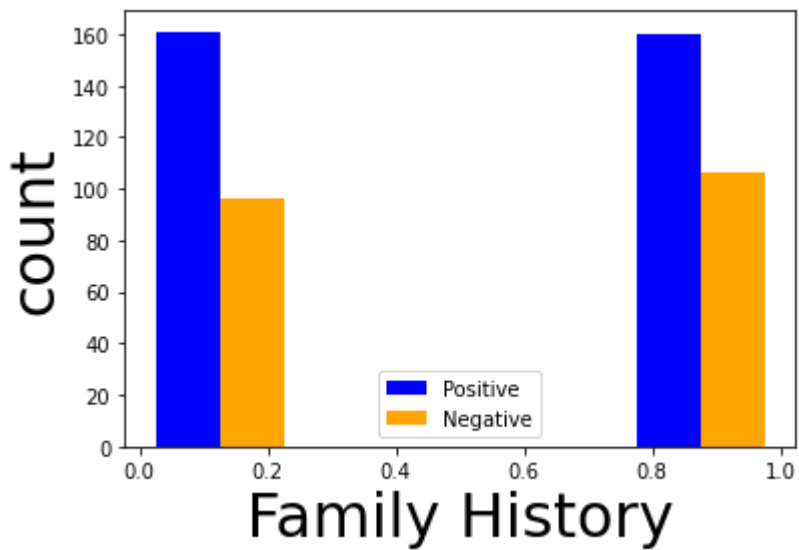
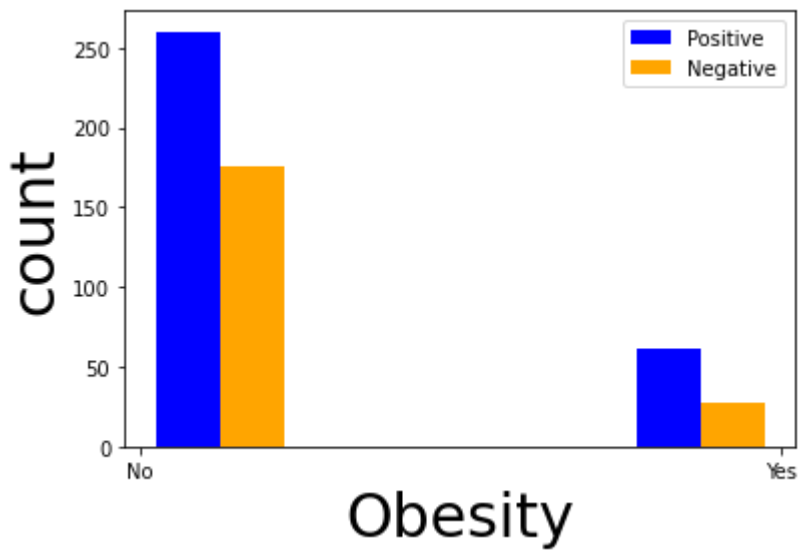






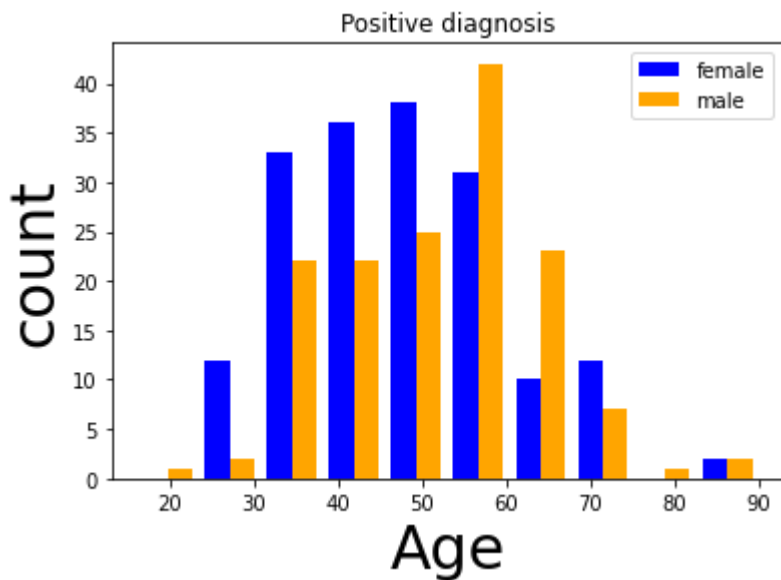







c. additional plots

```
In [7]: from DataExplorer import data_gender_explorer  
data_gender_explorer(T1D_dataset_clean)
```



d1. In the additional plot we can see that women in earlier age were diagnosed as 'Positive' more than men in the data. And as the age increases, men were diagnosed as 'Positive' more than women 

d2. Yes we have several features: Increased Urination, Increased Thirst, female in the Gender with strong correlation with the 'Positive' label.

sec 4

In this sec we encoded our data as one-hot vectors. instead of using sklearn function. we implement our own version which keeps the titles of the columns and gives a more understandable result. In addition, our function skips the 'Age' feature since this is a non-categorical feature and not suitable for one-hot encoding.

```
In [8]: from DataExplorer import encode_and_bind
one_hot_data = encode_and_bind(T1D_dataset_clean)
one_hot_data.head()
```

```
Out[8]:
```

	Age	Gender_Female	Gender_Male	Increased Urination_No	Increased Urination_Yes	Increased Thirst_No	Increased Thirst_Yes	Sudden Weight Loss_No	Sudden Weight Loss_Yes
0	45	0	1	1	0	1	0	1	
1	42	0	1	1	0	1	0	1	
2	45	0	1	0	1	0	1	1	
3	59	1	0	1	0	1	0	1	
4	40	1	0	0	1	0	1	0	

5 rows × 35 columns

We used our function only on the X_train and X_test. for the Y we transform it to numeric (1/0), but did not encode it as one-hot vector, Because we have come to the conclusion that it produces unnecessary duplication of data.

```
In [9]: X_train = encode_and_bind(X_train)
X_test = encode_and_bind(X_test)
X_test
```

```
Out[9]:
```

	Age	Gender_Female	Gender_Male	Increased Urination_No	Increased Urination_Yes	Increased Thirst_No	Increased Thirst_Yes	Sudden Weight Loss_No	Sudden Weight Loss_Yes
504	-0.018603	0	1	0	1	0	1		
146	-0.018603	1	0	0	1	0	1		
218	0.791409	1	0	0	1	1	0		
476	-0.747614	1	0	0	1	0	1		
286	-0.423609	1	0	0	1	0	1		

	Age	Gender_Female	Gender_Male	Increased Urination_No	Increased Urination_Yes	Increased Thirst_No	Increased Thirst_Yes	Sudden Weight Loss_No
...
347	0.386403	0	1	0	1	0	1	...
501	-0.909616	0	1	1	0	1	0	...
43	0.548405	1	0	0	1	0	1	...
464	0.386403	0	1	1	0	1	0	...
375	-0.423609	0	1	1	0	1	0	...

105 rows × 33 columns

```
In [10]: Y_train[Y_train=='Positive']=1
Y_train[Y_train=='Negative']=0
Y_test[Y_test=='Positive']=1
Y_test[Y_test=='Negative']=0
Y_train=Y_train.astype('int')
Y_test=Y_test.astype('int')
```



sec 5

5.a.1 Logistic Regression with k fold cross validation

```
In [11]: from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

max_iter = 2000
n_splits = 10
skf = StratifiedKFold(n_splits=n_splits, random_state=10, shuffle=True)
log_reg = LogisticRegression(random_state=5, max_iter=max_iter, solver='lbfgs')
lmbda = np.array([0.01, 0.01, 1, 10, 100, 1000])
pipe = Pipeline(steps=[('logistic', log_reg)])
clf = GridSearchCV(estimator=pipe, param_grid={'logistic_C': 1/lmbda, 'logistic_penalty': lmbda,
scoring=['accuracy', 'f1', 'precision', 'recall', 'roc_auc'], cv=skf,
refit='roc_auc', verbose=3, return_train_score=True)
clf.fit(X_train, Y_train)
print(clf.best_params_)
best_log_reg = clf.best_estimator_
```

Fitting 10 folds for each of 12 candidates, totalling 120 fits

[CV] logistic_C=100.0, logistic_penalty=12

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.5s

[CV] logistic_C=100.0, logistic_penalty=12

[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.949, test=0.905), f1=(tr

```

ain=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.2s
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.5s remaining: 0.0s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 0.7s remaining: 0.0s
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.4s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.1s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.938), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.1s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.949, test=0.952), f1=(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=0.960), roc_auc=(train=0.988, test=0.960), total= 0.1s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.939, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=l2 .....
[CV] logistic__C=100.0, logistic__penalty=l2, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=None, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=None, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=None, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.3s
[CV] logistic__C=100.0, logistic__penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=None, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.1s

```

```

=0.923), roc_auc=(train=0.986, test=0.995), total= 0.3s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.935), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.1s
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None .....
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.949, test=0.952), f1=(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=0.960), roc_auc=(train=0.988, test=0.960), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.939, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=1.000), roc_auc=(train=0.985, test=1.000), total= 0.3s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=12 .....
[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.4s
[CV] logistic_C=100.0, logistic_penalty=12 .....
[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.4s
[CV] logistic_C=100.0, logistic_penalty=12 .....
[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=12 .....
[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.3s
[CV] logistic_C=100.0, logistic_penalty=12 .....
[CV] logistic_C=100.0, logistic_penalty=12, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.938), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=12 .....

```

```

[CV] logistic_C=100.0, logistic_penalty=l2, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=l2 .....
[CV] logistic_C=100.0, logistic_penalty=l2, accuracy=(train=0.949, test=0.952), f1=(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=l2 .....
[CV] logistic_C=100.0, logistic_penalty=l2, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=0.960), roc_auc=(train=0.988, test=0.960), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=l2 .....
[CV] logistic_C=100.0, logistic_penalty=l2, accuracy=(train=0.939, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=l2 .....
[CV] logistic_C=100.0, logistic_penalty=l2, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=None .....
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.1s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.935), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=100.0, logistic_penalty=None, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.2s
[CV] logistic_C=100.0, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(

```

```

arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=none, accuracy=(train=0.949, test=0.952), f1=
(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test
=1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic__C=100.0, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=none, accuracy=(train=0.955, test=0.929), f1=
(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test
=0.960), roc_auc=(train=0.988, test=0.960), total= 0.5s
[CV] logistic__C=100.0, logistic__penalty=none .....
[CV] logistic__C=100.0, logistic__penalty=none, accuracy=(train=0.939, test=0.976), f1=
(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test
=1.000), roc_auc=(train=0.985, test=1.000), total= 0.1s
[CV] logistic__C=100.0, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=100.0, logistic__penalty=none, accuracy=(train=0.955, test=0.951), f1=
(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test
=0.960), roc_auc=(train=0.985, test=0.990), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.934, test=0.976), f1=(trai
n=0.946, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.939, test=0.96
2), roc_auc=(train=0.985, test=0.998), total= 0.2s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.929), f1=(trai
n=0.956, test=0.943), precision=(train=0.969, test=0.926), recall=(train=0.944, test=0.96
2), roc_auc=(train=0.986, test=0.988), total= 0.2s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.957, test=0.881), f1=(trai
n=0.965, test=0.898), precision=(train=0.969, test=0.957), recall=(train=0.961, test=0.84
6), roc_auc=(train=0.991, test=0.933), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.952, test=0.929), f1=(trai
n=0.961, test=0.941), precision=(train=0.965, test=0.960), recall=(train=0.957, test=0.92
3), roc_auc=(train=0.985, test=0.995), total= 0.2s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.905), f1=(trai
n=0.956, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.948, test=0.88
5), roc_auc=(train=0.989, test=0.945), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.929), f1=(trai
n=0.956, test=0.939), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.88
5), roc_auc=(train=0.987, test=0.971), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.944, test=0.952), f1=(trai
n=0.954, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.948, test=1.00
0), roc_auc=(train=0.986, test=1.000), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.947, test=0.905), f1=(trai
n=0.956, test=0.920), precision=(train=0.973, test=0.920), recall=(train=0.940, test=0.92
0), roc_auc=(train=0.987, test=0.979), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.936, test=1.000), f1=(trai
n=0.948, test=1.000), precision=(train=0.952, test=1.000), recall=(train=0.944, test=1.00
0), roc_auc=(train=0.985, test=1.000), total= 0.1s
[CV] logistic__C=1.0, logistic__penalty=l2 .....
[CV] logistic__C=1.0, logistic__penalty=l2, accuracy=(train=0.955, test=0.951), f1=(trai

```



```
n=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.988), total= 0.2s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.2s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.2s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.935), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.949, test=0.952), f1=(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=0.960), roc_auc=(train=0.988, test=0.960), total= 0.2s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.939, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=1.000), roc_auc=(train=0.985, test=1.000), total= 0.1s
[CV] logistic_C=1.0, logistic_penalty=None .....
[CV] logistic_C=1.0, logistic_penalty=None, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.923, test=0.929), f1=(train=0.936, test=0.939), precision=(train=0.951, test=1.000), recall=(train=0.922, test=0.885), roc_auc=(train=0.981, test=1.000), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.934, test=0.952), f1=(train=0.945, test=0.962), precision=(train=0.960, test=0.962), recall=(train=0.931, test=0.962), roc_auc=(train=0.983, test=0.988), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.944, test=0.833), f1=(train=0.954, test=0.851), precision=(train=0.965, test=0.952), recall=(train=0.944, test=0.769), roc_auc=(train=0.988, test=0.925), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.928, test=0.929), f1=(train=0.941, test=0.941), precision=(train=0.955, test=0.960), recall=(train=0.926, test=0.923), roc_auc=(train=0.982, test=0.995), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.936, test=0.833), f1=(train=0.947, test=0.857), precision=(train=0.968, test=0.913), recall=(train=0.926, test=0.808), roc_auc=(train=0.985, test=0.947), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.939, test=0.905), f1=(train=0.949, test=0.923), precision=(train=0.968, test=0.923), recall=(train=0.931, test=0.92
```

```

3), roc_auc=(train=0.983, test=0.978), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.918, test=0.952), f1=(train=0.932, test=0.963), precision=(train=0.942, test=0.929), recall=(train=0.922, test=1.000), roc_auc=(train=0.982, test=1.000), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.923, test=0.881), f1=(train=0.937, test=0.898), precision=(train=0.951, test=0.917), recall=(train=0.922, test=0.880), roc_auc=(train=0.983, test=0.979), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.920, test=0.951), f1=(train=0.935, test=0.962), precision=(train=0.943, test=0.926), recall=(train=0.927, test=1.000), roc_auc=(train=0.982, test=1.000), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=l2 .....
[CV] logistic_C=0.1, logistic_penalty=l2, accuracy=(train=0.915, test=0.927), f1=(train=0.930, test=0.941), precision=(train=0.946, test=0.923), recall=(train=0.914, test=0.960), roc_auc=(train=0.982, test=0.985), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.957, test=0.905), f1=(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=0.885), roc_auc=(train=0.992, test=0.935), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.947, test=0.929), f1=(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=0.923), roc_auc=(train=0.986, test=0.995), total= 0.2s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.949, test=0.881), f1=(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=0.885), roc_auc=(train=0.990, test=0.935), total= 0.4s
[CV] logistic_C=0.1, logistic_penalty=None .....
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.947, test=0.905), f1=(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=0.846), roc_auc=(train=0.987, test=0.962), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(

```

```

[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.949, test=0.952), f1=(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=1.000), roc_auc=(train=0.986, test=1.000), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=0.960), roc_auc=(train=0.988, test=0.960), total= 0.3s
[CV] logistic_C=0.1, logistic_penalty=None .....
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.939, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=1.000), roc_auc=(train=0.985, test=1.000), total= 0.1s
[CV] logistic_C=0.1, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.1, logistic_penalty=None, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.912, test=0.929), f1=(train=0.929, test=0.941), precision=(train=0.923, test=0.960), recall=(train=0.935, test=0.923), roc_auc=(train=0.968, test=0.988), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.912, test=0.952), f1=(train=0.929, test=0.962), precision=(train=0.923, test=0.962), recall=(train=0.935, test=0.962), roc_auc=(train=0.967, test=0.986), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.920, test=0.833), f1=(train=0.936, test=0.857), precision=(train=0.928, test=0.913), recall=(train=0.944, test=0.808), roc_auc=(train=0.975, test=0.930), total= 0.0s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.912, test=0.905), f1=(train=0.929, test=0.920), precision=(train=0.923, test=0.958), recall=(train=0.935, test=0.885), roc_auc=(train=0.967, test=0.981), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.920, test=0.810), f1=(train=0.936, test=0.833), precision=(train=0.928, test=0.909), recall=(train=0.944, test=0.769), roc_auc=(train=0.971, test=0.909), total= 0.0s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.915, test=0.881), f1=(train=0.931, test=0.906), precision=(train=0.923, test=0.889), recall=(train=0.939, test=0.923), roc_auc=(train=0.971, test=0.971), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.918, test=0.929), f1=(train=0.933, test=0.945), precision=(train=0.931, test=0.897), recall=(train=0.935, test=1.000), roc_auc=(train=0.968, test=0.990), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.912, test=0.929), f1=(train=0.929, test=0.941), precision=(train=0.927, test=0.923), recall=(train=0.931, test=0.960), roc_auc=(train=0.968, test=0.962), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.915, test=0.927), f1=(train=0.931, test=0.943), precision=(train=0.927, test=0.893), recall=(train=0.935, test=1.000), roc_auc=(train=0.966, test=0.985), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=l2 .....
[CV] logistic_C=0.01, logistic_penalty=l2, accuracy=(train=0.912, test=0.951), f1=(train=0.929, test=0.962), precision=(train=0.927, test=0.926), recall=(train=0.931, test=1.000), roc_auc=(train=0.969, test=0.973), total= 0.1s

```

```
[CV] logistic_C=0.01, logistic_penalty=None .....
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.936, test=0.976), f1=(t
rain=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=
0.962), roc_auc=(train=0.985, test=0.998), total= 0.1s
[CV] logistic_C=0.01, logistic_penalty=None .....
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.949, test=0.905), f1=(t
rain=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=
0.962), roc_auc=(train=0.986, test=0.983), total= 0.1s
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.957, test=0.905), f1=(t
rain=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test=
0.885), roc_auc=(train=0.992, test=0.935), total= 0.3s
[CV] logistic_C=0.01, logistic_penalty=None .....
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.947, test=0.929), f1=(t
rain=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test=
0.923), roc_auc=(train=0.986, test=0.995), total= 0.2s
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None .....
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.949, test=0.881), f1=(t
rain=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test=
0.885), roc_auc=(train=0.990, test=0.935), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.947, test=0.905), f1=(t
rain=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test=
0.846), roc_auc=(train=0.987, test=0.962), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.949, test=0.952), f1=(t
rain=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test=
1.000), roc_auc=(train=0.986, test=1.000), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.955, test=0.929), f1=(t
rain=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test=
0.960), roc_auc=(train=0.988, test=0.960), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.939, test=0.976), f1=(t
rain=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test=
```

```

1.000), roc_auc=(train=0.985, test=1.000), total= 0.2s
[CV] logistic_C=0.01, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.01, logistic_penalty=None, accuracy=(train=0.955, test=0.951), f1=(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.960), roc_auc=(train=0.985, test=0.990), total= 0.2s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.948, test=0.969), total= 0.2s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.946, test=0.971), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.955, test=0.923), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.946, test=0.964), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.954, test=0.868), total= 0.0s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.952, test=0.969), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.948, test=0.971), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.617, test=0.595), f1=(train=0.763, test=0.746), precision=(train=0.617, test=0.595), recall=(train=1.000, test=1.000), roc_auc=(train=0.950, test=0.927), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.615, test=0.610), f1=(train=0.762, test=0.758), precision=(train=0.615, test=0.610), recall=(train=1.000, test=1.000), roc_auc=(train=0.946, test=0.968), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=l2 .....
[CV] logistic_C=0.001, logistic_penalty=l2, accuracy=(train=0.615, test=0.610), f1=(train=0.762, test=0.758), precision=(train=0.615, test=0.610), recall=(train=1.000, test=1.000), roc_auc=(train=0.948, test=0.963), total= 0.1s
[CV] logistic_C=0.001, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(
[CV] logistic_C=0.001, logistic_penalty=None, accuracy=(train=0.936, test=0.976), f1=(train=0.948, test=0.980), precision=(train=0.952, test=1.000), recall=(train=0.944, test=0.962), roc_auc=(train=0.985, test=0.998), total= 0.2s
[CV] logistic_C=0.001, logistic_penalty=None .....
[CV] logistic_C=0.001, logistic_penalty=None, accuracy=(train=0.949, test=0.905), f1=(train=0.958, test=0.926), precision=(train=0.969, test=0.893), recall=(train=0.948, test=0.962), roc_auc=(train=0.986, test=0.983), total= 0.2s
[CV] logistic_C=0.001, logistic_penalty=None .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserWarning: Setting penalty='none' will ignore the C and l1_ratio parameters
  warnings.warn(

```

```

arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.957, test=0.905), f1=
(train=0.965, test=0.920), precision=(train=0.965, test=0.958), recall=(train=0.965, test
=0.885), roc_auc=(train=0.992, test=0.935), total= 0.2s
[CV] logistic__C=0.001, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.947, test=0.929), f1=
(train=0.957, test=0.941), precision=(train=0.957, test=0.960), recall=(train=0.957, test
=0.923), roc_auc=(train=0.986, test=0.995), total= 0.2s
[CV] logistic__C=0.001, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.949, test=0.881), f1=
(train=0.959, test=0.902), precision=(train=0.961, test=0.920), recall=(train=0.957, test
=0.885), roc_auc=(train=0.990, test=0.935), total= 0.4s
[CV] logistic__C=0.001, logistic__penalty=none .....
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.947, test=0.905), f1=
(train=0.956, test=0.917), precision=(train=0.965, test=1.000), recall=(train=0.948, test
=0.846), roc_auc=(train=0.987, test=0.962), total= 0.1s
[CV] logistic__C=0.001, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.949, test=0.952), f1=
(train=0.959, test=0.963), precision=(train=0.961, test=0.929), recall=(train=0.957, test
=1.000), roc_auc=(train=0.986, test=1.000), total= 0.1s
[CV] logistic__C=0.001, logistic__penalty=none .....
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.955, test=0.929), f1=
(train=0.963, test=0.941), precision=(train=0.978, test=0.923), recall=(train=0.948, test
=0.960), roc_auc=(train=0.988, test=0.960), total= 0.3s
[CV] logistic__C=0.001, logistic__penalty=none .....
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.939, test=0.976), f1=
(train=0.950, test=0.980), precision=(train=0.952, test=0.962), recall=(train=0.948, test
=1.000), roc_auc=(train=0.985, test=1.000), total= 0.1s
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
C:\Users\Elinoy\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:1320: UserW
arning: Setting penalty='none' will ignore the C and l1_ratio parameters
warnings.warn(
[CV] logistic__C=0.001, logistic__penalty=none .....
[CV] logistic__C=0.001, logistic__penalty=none, accuracy=(train=0.955, test=0.951), f1=
(train=0.963, test=0.960), precision=(train=0.978, test=0.960), recall=(train=0.948, test
=0.960), roc_auc=(train=0.985, test=0.990), total= 0.1s
[Parallel(n_jobs=1)]: Done 120 out of 120 | elapsed: 25.5s finished
{'logistic__C': 0.1, 'logistic__penalty': 'l2'}

```

5.a.2 non-linear svm with k fold cross validation

```

In [12]: from sklearn.svm import SVC
svc = SVC(probability=True)
C = np.array([0.001, 0.01, 1, 10, 100, 1000])

```

```

pipe = Pipeline(steps=[('svm', svc)])
svm_nonlin = GridSearchCV(estimator=pipe,
                           param_grid={'svm__kernel':['rbf','poly'], 'svm__C':C, 'svm__degree':[3], 's
                           scoring=['accuracy','f1','precision','recall','roc_auc'],
                           cv=skf, refit='roc_auc', verbose=3, return_train_score=True)
svm_nonlin.fit(X_train, Y_train)
best_svm_nonlin = svm_nonlin.best_estimator_
print(svm_nonlin.best_params_)

```

Fitting 10 folds for each of 24 candidates, totalling 240 fits

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.962, test=0.978), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.2s remaining: 0.0s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.960, test=0.974), total= 0.3s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 0.6s remaining: 0.0s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.966, test=0.921), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.960, test=0.978), total= 0.3s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.966, test=0.897), total= 0.1s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.963, test=0.966), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.962, test=0.981), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.617, test=0.595), f1=(train=0.763, test=0.746), precision=(train=0.617, test=0.595), recall=(train=1.000, test=1.000), roc_auc=(train=0.964, test=0.951), total= 0.1s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.615, test=0.610), f1=(train=0.762, test=0.758), precision=(train=0.615, test=0.610), recall=(train=1.000, test=1.000), roc_auc=(train=0.959, test=0.982), total= 0.1s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf ...

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=rbf, accuracy=(train=0.615, test=0.610), f1=(train=0.762, test=0.758), precision=(train=0.615, test=0.610), recall=(train=1.000, test=1.000), roc_auc=(train=0.962, test=0.975), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly ..

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.970, test=0.971), total= 0.1s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly ..

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly, accuracy=(train=0.614, test=0.619), f1=(train=0.761, test=0.765), precision=(train=0.614, test=0.619), recall=(train=1.000, test=1.000), roc_auc=(train=0.967, test=0.986), total= 0.2s

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly ..

[CV] svm__C=0.001, svm__degree=3, svm__gamma=auto, svm__kernel=poly, accuracy=(train=0.6

24/43

[illegible]

26/43

[illegible]

```

31, test=0.952), f1=(train=0.944, test=0.962), precision=(train=0.940, test=0.926), recall=(train=0.948, test=1.000), roc_auc=(train=0.979, test=0.984), total= 0.1s
[CV] svm_C=0.01, svm_degree=3, svm_gamma=scale, svm_kernel=poly ..
[CV] svm_C=0.01, svm_degree=3, svm_gamma=scale, svm_kernel=poly, accuracy=(train=0.936, test=0.927), f1=(train=0.948, test=0.943), precision=(train=0.944, test=0.893), recall=(train=0.953, test=1.000), roc_auc=(train=0.979, test=0.998), total= 0.2s
[CV] svm_C=0.01, svm_degree=3, svm_gamma=scale, svm_kernel=poly ..
[CV] svm_C=0.01, svm_degree=3, svm_gamma=scale, svm_kernel=poly, accuracy=(train=0.931, test=0.951), f1=(train=0.944, test=0.962), precision=(train=0.940, test=0.926), recall=(train=0.948, test=1.000), roc_auc=(train=0.980, test=0.985), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.949, test=0.976), f1=(train=0.958, test=0.980), precision=(train=0.977, test=1.000), recall=(train=0.939, test=0.962), roc_auc=(train=0.991, test=1.000), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.960, test=0.929), f1=(train=0.967, test=0.943), precision=(train=0.986, test=0.926), recall=(train=0.948, test=0.962), roc_auc=(train=0.993, test=0.988), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.957, test=0.881), f1=(train=0.965, test=0.894), precision=(train=0.986, test=1.000), recall=(train=0.944, test=0.808), roc_auc=(train=0.995, test=0.959), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.952, test=0.952), f1=(train=0.960, test=0.960), precision=(train=0.978, test=1.000), recall=(train=0.944, test=0.923), roc_auc=(train=0.991, test=0.995), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.960, test=0.857), f1=(train=0.967, test=0.875), precision=(train=0.982, test=0.955), recall=(train=0.952, test=0.808), roc_auc=(train=0.993, test=0.974), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.955, test=0.929), f1=(train=0.963, test=0.941), precision=(train=0.978, test=0.960), recall=(train=0.948, test=0.923), roc_auc=(train=0.992, test=0.990), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.947, test=0.976), f1=(train=0.956, test=0.981), precision=(train=0.965, test=0.963), recall=(train=0.948, test=1.000), roc_auc=(train=0.991, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.952, test=0.905), f1=(train=0.960, test=0.917), precision=(train=0.982, test=0.957), recall=(train=0.940, test=0.880), roc_auc=(train=0.992, test=0.981), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.955, test=1.000), f1=(train=0.963, test=1.000), precision=(train=0.969, test=1.000), recall=(train=0.957, test=1.000), roc_auc=(train=0.990, test=1.000), total= 0.4s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf .....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=rbf, accuracy=(train=0.955, test=0.976), f1=(train=0.963, test=0.980), precision=(train=0.982, test=0.962), recall=(train=0.944, test=1.000), roc_auc=(train=0.991, test=0.988), total= 0.3s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.941, test=0.952), f1=(train=0.952, test=0.960), precision=(train=0.964, test=1.000), recall=(train=0.939, test=0.923), roc_auc=(train=0.989, test=0.995), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.947, test=0.976), f1=(train=0.956, test=0.981), precision=(train=0.965, test=0.963), recall=(train=0.948, test=1.000), roc_auc=(train=0.991, test=0.993), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.955, test=0.857), f1=(train=0.963, test=0.875), precision=(train=0.969, test=0.955), recall=(train=0.957, test=0.808), roc_auc=(train=0.993, test=0.950), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.949, test=0.929), f1=(train=0.958, test=0.941), precision=(train=0.969, test=0.960), recall=

```

```

=(train=0.948, test=0.923), roc_auc=(train=0.988, test=0.986), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.94
7, test=0.833), f1=(train=0.956, test=0.851), precision=(train=0.965, test=0.952), recall
=(train=0.948, test=0.769), roc_auc=(train=0.990, test=0.978), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.95
2, test=0.905), f1=(train=0.961, test=0.923), precision=(train=0.973, test=0.923), recall
=(train=0.948, test=0.923), roc_auc=(train=0.990, test=0.983), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.94
4, test=0.929), f1=(train=0.954, test=0.945), precision=(train=0.961, test=0.897), recall
=(train=0.948, test=1.000), roc_auc=(train=0.989, test=1.000), total= 0.3s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.93
9, test=0.929), f1=(train=0.950, test=0.939), precision=(train=0.952, test=0.958), recall
=(train=0.948, test=0.920), roc_auc=(train=0.989, test=0.995), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.94
4, test=0.976), f1=(train=0.954, test=0.980), precision=(train=0.961, test=0.962), recall
=(train=0.948, test=1.000), roc_auc=(train=0.988, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=auto, svm_kernel=poly, accuracy=(train=0.93
9, test=0.976), f1=(train=0.950, test=0.980), precision=(train=0.964, test=0.962), recall
=(train=0.935, test=1.000), roc_auc=(train=0.991, test=0.983), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
9, test=1.000), f1=(train=0.991, test=1.000), precision=(train=0.991, test=1.000), recall
=(train=0.991, test=1.000), roc_auc=(train=1.000, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
9, test=0.929), f1=(train=0.991, test=0.943), precision=(train=0.987, test=0.926), recall
=(train=0.996, test=0.962), roc_auc=(train=0.999, test=0.990), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.99
2, test=0.929), f1=(train=0.994, test=0.939), precision=(train=0.991, test=1.000), recall
=(train=0.996, test=0.885), roc_auc=(train=1.000, test=0.978), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
9, test=0.976), f1=(train=0.991, test=0.980), precision=(train=0.987, test=1.000), recall
=(train=0.996, test=0.962), roc_auc=(train=0.999, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.99
2, test=0.929), f1=(train=0.994, test=0.941), precision=(train=0.991, test=0.960), recall
=(train=0.996, test=0.923), roc_auc=(train=0.999, test=0.983), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
7, test=0.976), f1=(train=0.989, test=0.981), precision=(train=0.987, test=0.963), recall
=(train=0.991, test=1.000), roc_auc=(train=0.999, test=0.995), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
7, test=1.000), f1=(train=0.989, test=1.000), precision=(train=0.987, test=1.000), recall
=(train=0.991, test=1.000), roc_auc=(train=0.999, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
9, test=0.952), f1=(train=0.991, test=0.960), precision=(train=0.991, test=0.960), recall
=(train=0.991, test=0.960), roc_auc=(train=0.999, test=0.986), total= 0.1s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
9, test=1.000), f1=(train=0.991, test=1.000), precision=(train=0.987, test=1.000), recall
=(train=0.996, test=1.000), roc_auc=(train=0.999, test=1.000), total= 0.2s
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf ....
[CV] svm_C=1.0, svm_degree=3, svm_gamma=scale, svm_kernel=rbf, accuracy=(train=0.98
4, test=0.951), f1=(train=0.987, test=0.960), precision=(train=0.983, test=0.960), recall
=(train=0.991, test=0.960), roc_auc=(train=0.999, test=0.998), total= 0.1s

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
call=(train=1.000, test=1.000), roc_auc=(train=1.000, test=0.998), total= 0.1s
{'svm__C': 1.0, 'svm__degree': 3, 'svm__gamma': 'scale', 'svm__kernel': 'rbf'}
[Parallel(n_jobs=1)]: Done 240 out of 240 | elapsed: 52.5s finished
```

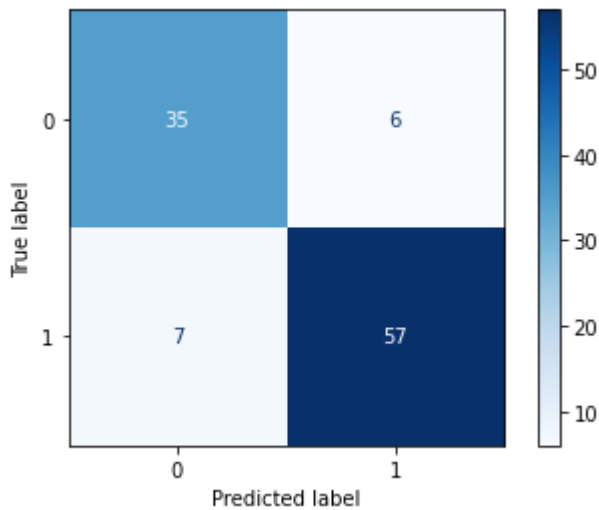
5.b evaluation metrics for both model :

```
In [13]: from sklearn.metrics import log_loss
best_log_reg.fit(X_train,Y_train)
y_pred_train = best_log_reg.predict_proba(X_train)
y_pred_test = best_log_reg.predict_proba(X_test)
print("train loss is {:.2f}".format(log_loss(Y_train,y_pred_train)))
print("test loss is {:.2f}".format(log_loss(Y_test,y_pred_test)))
```

train loss is 0.22
test loss is 0.29



```
In [14]: from Classifier import plot_evel
plot_evel(best_log_reg,X_test,Y_test)
```

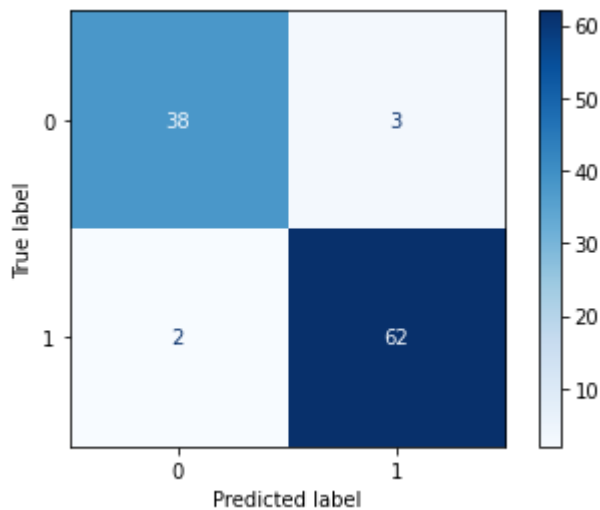


Sensitivity is 0.89. Specificity is 0.85. PPV is 0.90. NPV is 0.83. Accuracy is 0.88. F1 is 0.90.
AUROC is 0.94

```
In [15]: best_svm_nonlin.fit(X_train,Y_train)
```


```
Out[15]: Pipeline(steps=[('svm', SVC(probability=True))])
```

```
In [16]: plot_evel(best_svm_nonlin,X_test,Y_test)
```



Sensitivity is 0.97. Specificity is 0.93. PPV is 0.95. NPV is 0.95. Accuracy is 0.95. F1 is 0.96.

AUROC is 0.98

5.c the non-linear model performed better. 

sec 6

```
In [17]: from sklearn.ensemble import RandomForestClassifier as rfc
import matplotlib
#Load boston housing dataset as an example
names = list(X_train.columns)
clf = rfc(n_estimators=10)
clf.fit(X_train, Y_train)
```

Out[17]: RandomForestClassifier(n_estimators=10)

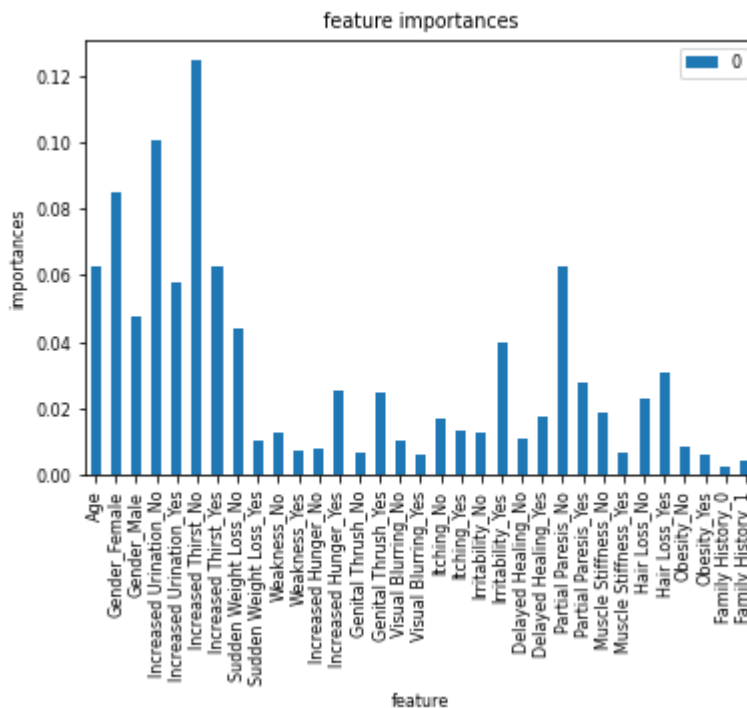
```
In [18]: plt.figure(figsize=(14, 10))
matplotlib.rcParams.update({'font.size': 8})
feature_importances = pd.DataFrame(clf.feature_importances_)

ax = feature_importances.plot(kind='bar')
ax.set_title('feature importances')
ax.set_xlabel('feature')
ax.set_ylabel('importances')
ax.set_xticklabels(names)
```

```
Out[18]: [Text(0, 0, 'Age'),
Text(0, 0, 'Gender_Female'),
Text(0, 0, 'Gender_Male'),
Text(0, 0, 'Increased Urination_No'),
Text(0, 0, 'Increased Urination_Yes'),
Text(0, 0, 'Increased Thirst_No'),
Text(0, 0, 'Increased Thirst_Yes'),
Text(0, 0, 'Sudden Weight Loss_No'),
Text(0, 0, 'Sudden Weight Loss_Yes'),
Text(0, 0, 'Weakness_No'),
Text(0, 0, 'Weakness_Yes'),
Text(0, 0, 'Increased Hunger_No'),
Text(0, 0, 'Increased Hunger_Yes'),
Text(0, 0, 'Genital Thrush_No'),
```

```
Text(0, 0, 'Genital Thrush_Yes'),
Text(0, 0, 'Visual Blurring_No'),
Text(0, 0, 'Visual Blurring_Yes'),
Text(0, 0, 'Itching_No'),
Text(0, 0, 'Itching_Yes'),
Text(0, 0, 'Irritability_No'),
Text(0, 0, 'Irritability_Yes'),
Text(0, 0, 'Delayed Healing_No'),
Text(0, 0, 'Delayed Healing_Yes'),
Text(0, 0, 'Partial Paresis_No'),
Text(0, 0, 'Partial Paresis_Yes'),
Text(0, 0, 'Muscle Stiffness_No'),
Text(0, 0, 'Muscle Stiffness_Yes'),
Text(0, 0, 'Hair Loss_No'),
Text(0, 0, 'Hair Loss_Yes'),
Text(0, 0, 'Obesity_No'),
Text(0, 0, 'Obesity_Yes'),
Text(0, 0, 'Family History_0'),
Text(0, 0, 'Family History_1')]
```

<Figure size 1008x720 with 0 Axes>



6.a.1 the most important features are increased thirst and increased urination

6.a.2 Yes, these results match up exactly with the feature exploration we did.

sec 7

a. dimensionality reduction with PCA

In [21]:

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scale = scaler.fit_transform(X_train)
X_test_scale = scaler.transform(X_test)
n_components = 2
pca = PCA(n_components=n_components, whiten=True)
```



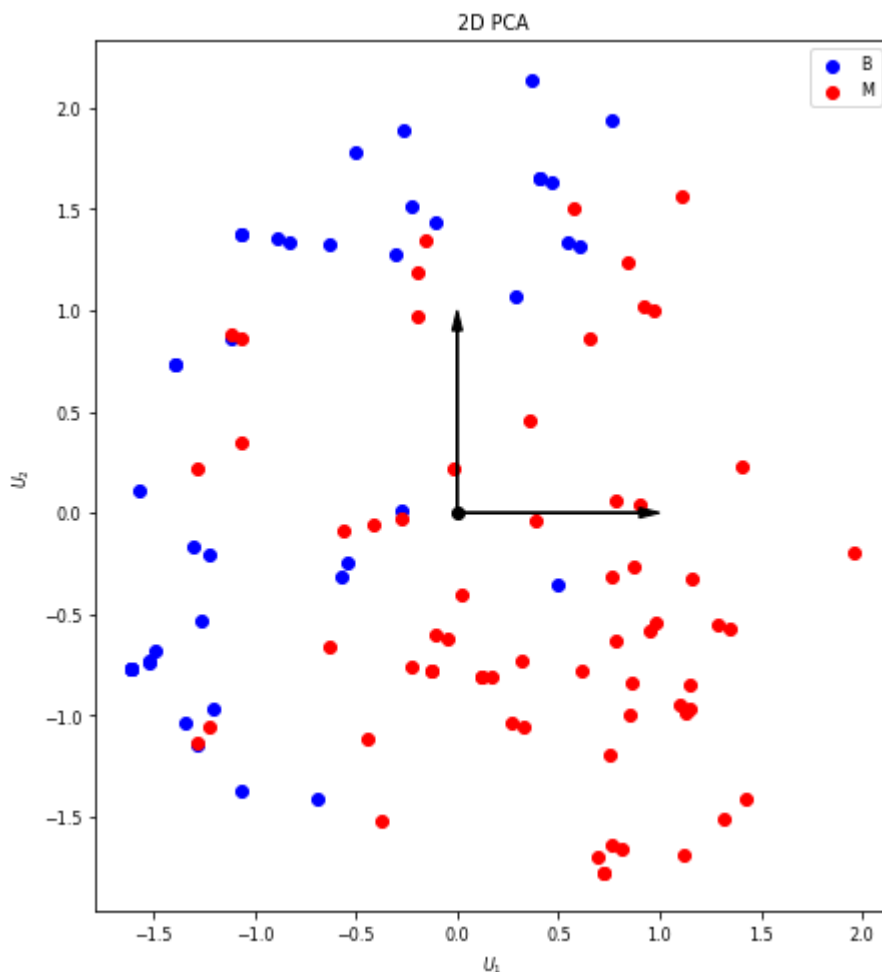
```
X_train_pca = pca.fit_transform(X_train_scale)
X_test_pca = pca.transform(X_test_scale)
```

In [22]:

```
def plt_2d_pca(X_pca,y):
    fig = plt.figure(figsize=(8, 8))
    ax = fig.add_subplot(111, aspect='equal')
    ax.scatter(X_pca[y==0, 0], X_pca[y==0, 1], color='b')
    ax.scatter(X_pca[y==1, 0], X_pca[y==1, 1], color='r')
    ax.legend(('B', 'M'))
    ax.plot([0], [0], "ko")
    ax.arrow(0, 0, 0, 1, head_width=0.05, length_includes_head=True, head_length=0.1, fc)
    ax.arrow(0, 0, 1, 0, head_width=0.05, length_includes_head=True, head_length=0.1, fc)
    ax.set_xlabel('$U_1$')
    ax.set_ylabel('$U_2$')
    ax.set_title('2D PCA')
```

In [23]:

```
plt_2d_pca(X_test_pca,Y_test)
```



b. Overall we can distinguish between area were each label is more common. However there are areas where data separation is not so straightforward

C. trainig non linear svm and logistic regression with dimension reduced dataset

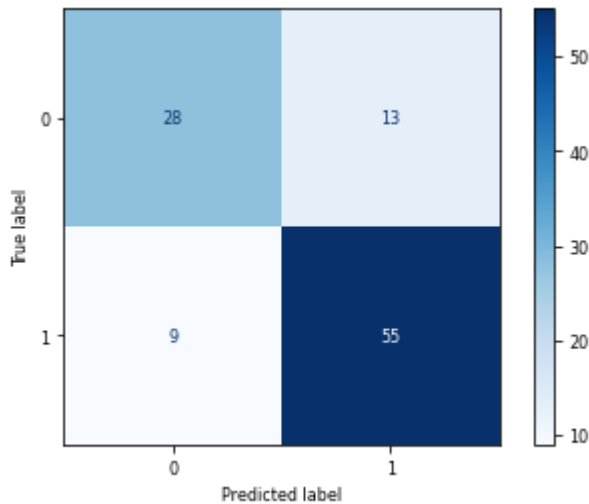
In [24]:

```
pipe_pca = Pipeline(steps=[('scale', StandardScaler()), ('pca', pca), ('logistic', best_l)
pipe_pca.fit(X_train, Y_train)
```

```
print('The score on the test set with PCA preprocessing is {:.2f}'.format(pipe_pca.score
```

The score on the test set with PCA preprocessing is 0.79

```
In [25]: plot_evel(pipe_pca,X_test,Y_test)
```

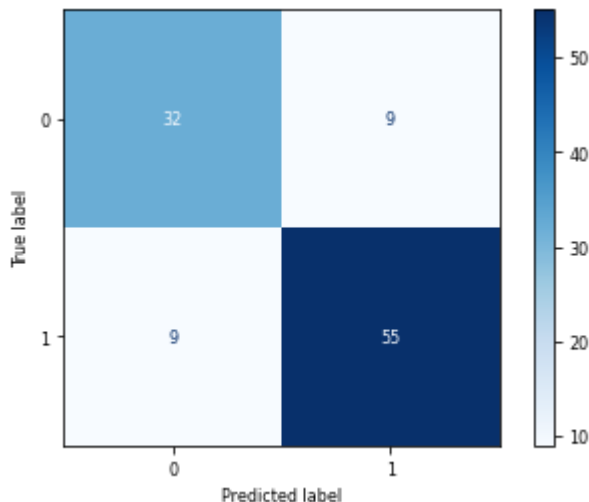


Sensitivity is 0.86. Specificity is 0.68. PPV is 0.81. NPV is 0.76. Accuracy is 0.79. F1 is 0.83.
AUROC is 0.90

```
In [26]: pipe_pca_svm = Pipeline(steps=[('scale', StandardScaler()),('pca', pca), ('svm', best_svm)])
pipe_pca_svm.fit(X_train, Y_train)
print('The score on the test set with PCA preprocessing is {:.2f}'.format(pipe_pca_svm.s
```

The score on the test set with PCA preprocessing is 0.83

```
In [27]: plot_evel(pipe_pca_svm,X_test,Y_test)
```



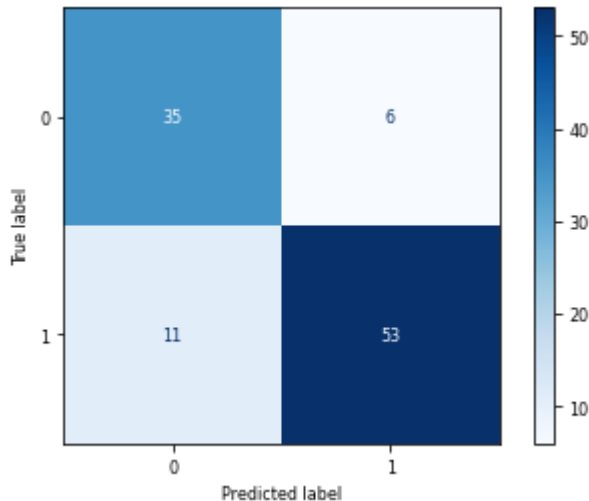
Sensitivity is 0.86. Specificity is 0.78. PPV is 0.86. NPV is 0.78. Accuracy is 0.83. F1 is 0.86.
AUROC is 0.93

d. training non linear svm and logistic regression on reduced data set which include only the two most important feat as was calculated in sec 6

```
In [28]:
```

```
cols = ['Increased_Thirst_Yes', 'Increased_Urination_Yes']
X_train_two_feat = X_train[cols]
X_test_two_feat = X_test[cols]
```

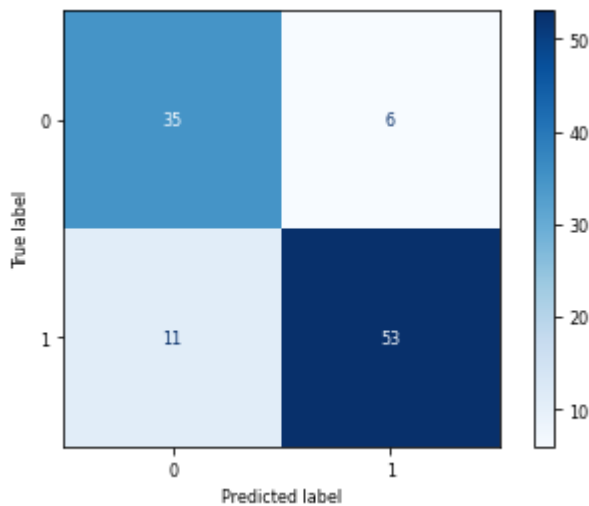
```
In [29]: best_log_reg.fit(X_train_two_feat,Y_train)
plot_evel(best_log_reg,X_test_two_feat,Y_test)
```



Sensitivity is 0.83. Specificity is 0.85. PPV is 0.90. NPV is 0.76. Accuracy is 0.84. F1 is 0.86.

AUROC is 0.88

```
In [30]: best_svm_nonlin.fit(X_train_two_feat,Y_train)
plot_evel(best_svm_nonlin,X_test_two_feat,Y_test)
```



Sensitivity is 0.83. Specificity is 0.85. PPV is 0.90. NPV is 0.76. Accuracy is 0.84. F1 is 0.86.

AUROC is 0.83

e. The preformems of all models are similar. The PCA preform better with the both model according to AUROC. the PCA SVM non-linear model preformem a little bit better then the linear model and on the 2 feat they give the same results.