Abstract

Introduction

Data imbalance poses a significant challenge in machine learning tasks, especially in classification problems. In this project, various resampling methods (SMOTE, ADASYN, NearMiss, Cluster-Based) were employed to address this issue. An ensemble model was also implemented to leverage the strengths of individual classifiers. The goal was to improve classification performance, particularly for the minority class.

Methodology and Results

Artificial Neural Network

Artificial Neural Network (ANN) Architectures

1 Hidden Layer ANN:

- **Input Layer**: Contains 21 neurons representing the features of the dataset.
- **Hidden Layer**: A single hidden layer with 16 neurons, using the ReLU activation function.
- **Output Layer**: Contains 2 neurons, corresponding to the number of classes in the label data. The activation function is softmax to output class probabilities.

2 Hidden Layers ANN:

- **Total Layers**: 4 layers in total.
- **Input Layer**: Contains 21 neurons.
- **Hidden Layers**:
  - The first hidden layer has 16 neurons with ReLU activation.
  - The second hidden layer has 8 neurons, also using ReLU activation.
- **Output Layer**: Similar to the single-hidden-layer ANN, this layer has 2 neurons with a softmax activation function.

Dropout Layers and Their Purpose

Dropout layers are set after the hidden layers in both architectures, with a dropout rate of 50%.With this approach, we aimed to prevent the models from overfitting. By introducing dropout, the models were less likely to memorize the training data and more likely to perform better on unseen data.

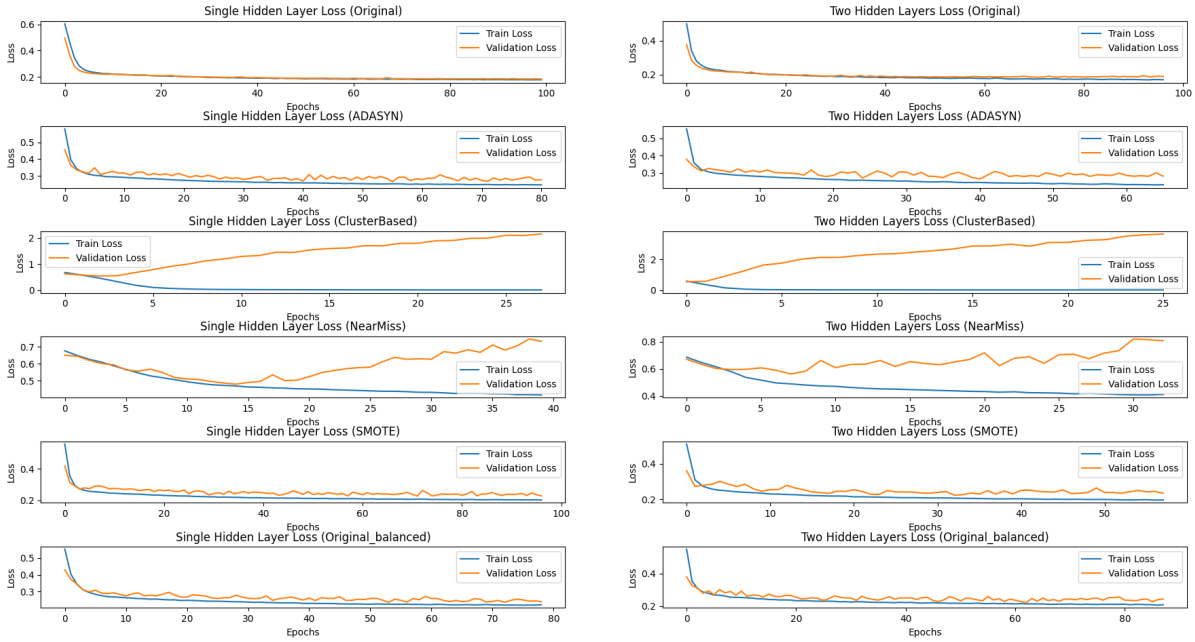
Training Strategy and Early Stopping

During the training phase:

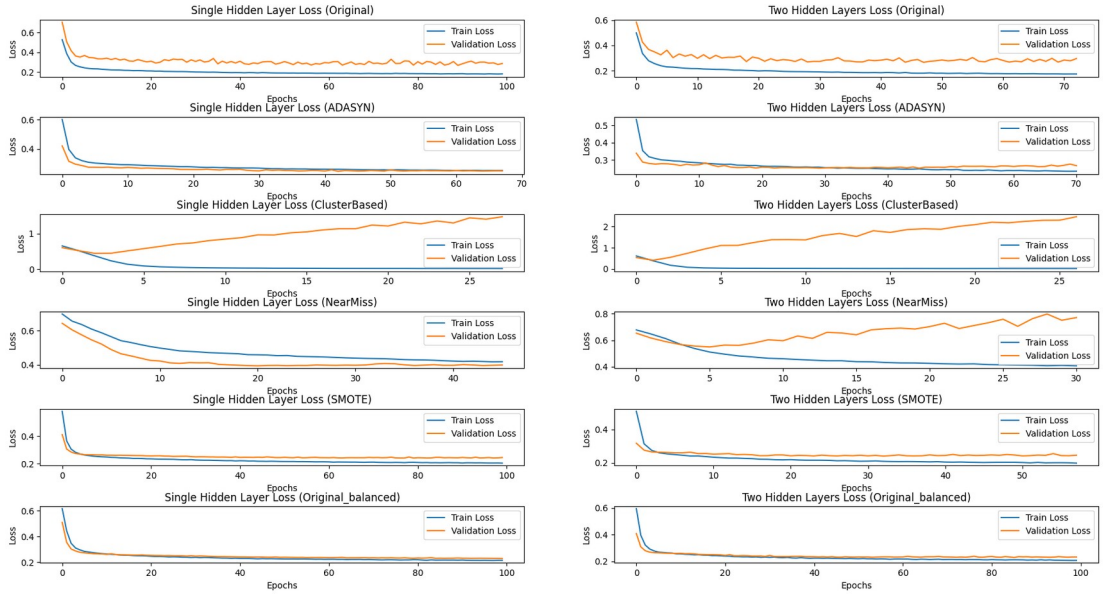
1. **Early Stopping**: Validation loss is monitored to prevent overfitting. If no improvement in validation loss is observed for 25 consecutive epochs, training is halted early.
2. **Checkpointing**: The model's parameters are saved at the epoch where the validation loss is at its minimum, ensuring that the best-performing model is retained.

With this approach, we tried to avoid overfitting and underfitting.

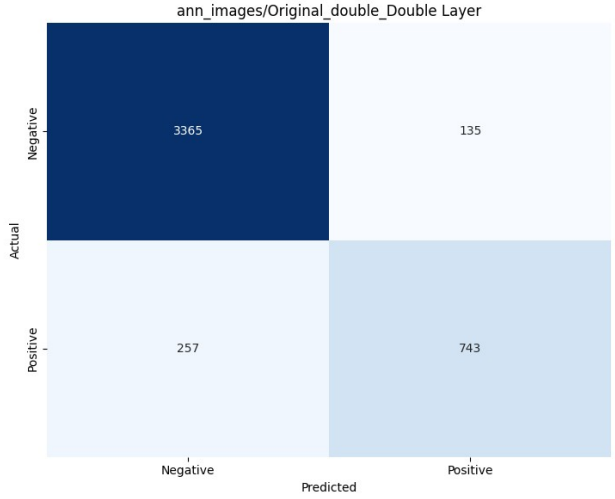
Training and validation loss graphs for every model:



Training and validation loss graphs for every model for undersampled validation test:

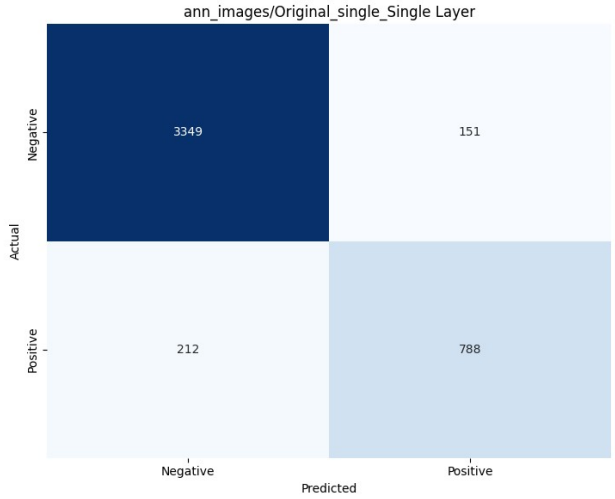


Original Data Set Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



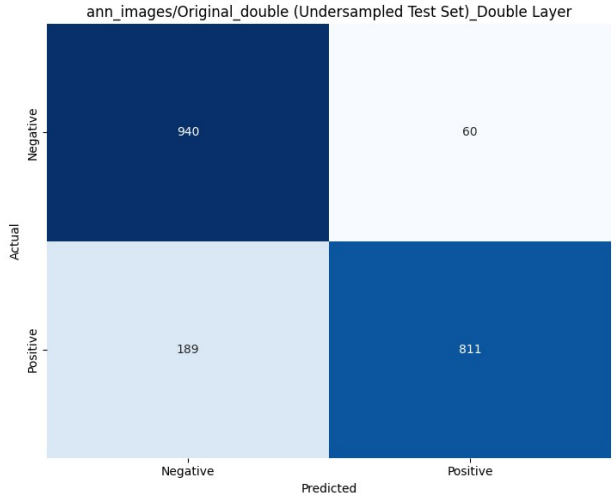
Metrics

Accuracy: 0.91
Precision: 0.85
Recall (Sensitivity): 0.74
Specificity: 0.96
F1 Score: 0.79



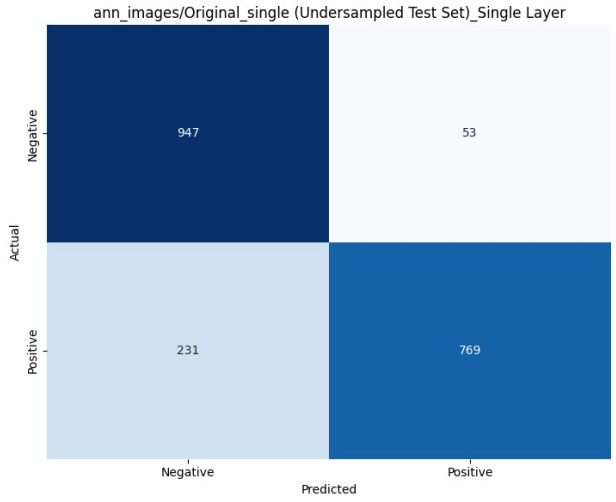
Metrics

Accuracy: 0.92
Precision: 0.84
Recall (Sensitivity): 0.79
Specificity: 0.96
F1 Score: 0.81



Metrics

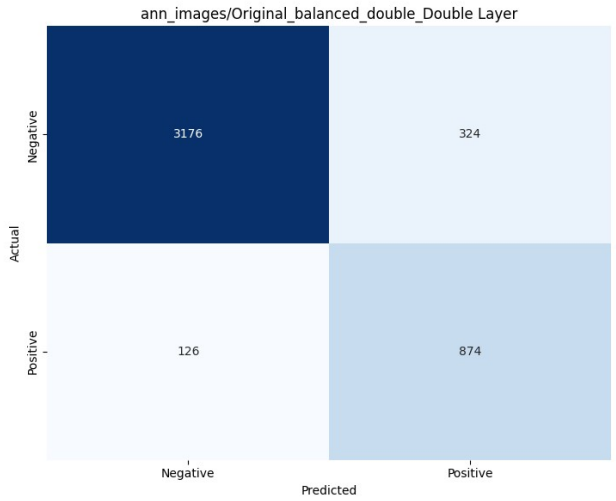
Accuracy: 0.88
Precision: 0.93
Recall (Sensitivity): 0.81
Specificity: 0.94
F1 Score: 0.87



Metrics

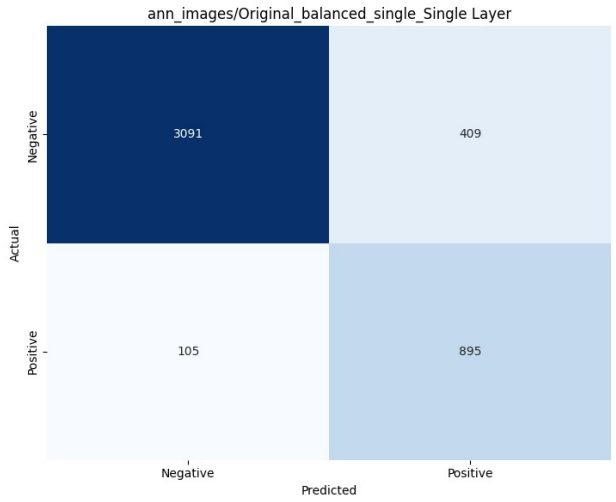
Accuracy: 0.86
Precision: 0.94
Recall (Sensitivity): 0.77
Specificity: 0.95
F1 Score: 0.84

Original Data Set with weight assigned Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



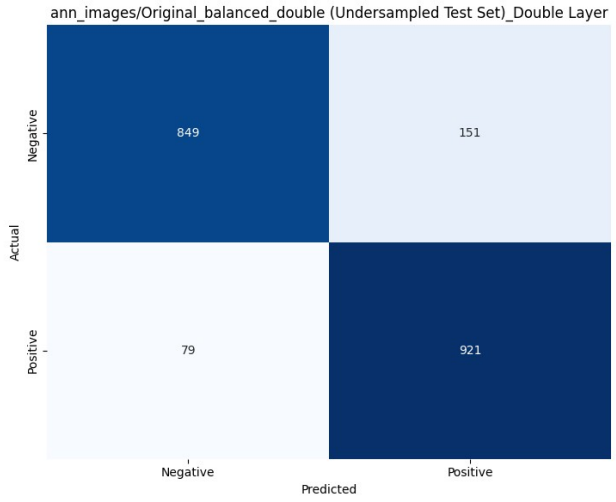
Metrics

Accuracy: 0.90
Precision: 0.73
Recall (Sensitivity): 0.87
Specificity: 0.91
F1 Score: 0.80



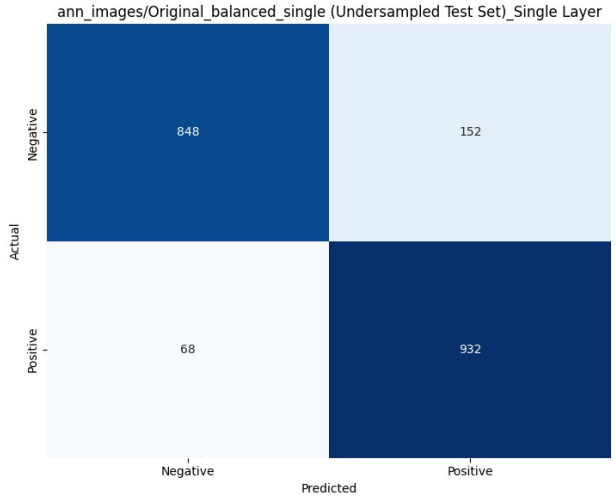
Metrics

Accuracy: 0.89
Precision: 0.69
Recall (Sensitivity): 0.90
Specificity: 0.88
F1 Score: 0.78



Metrics

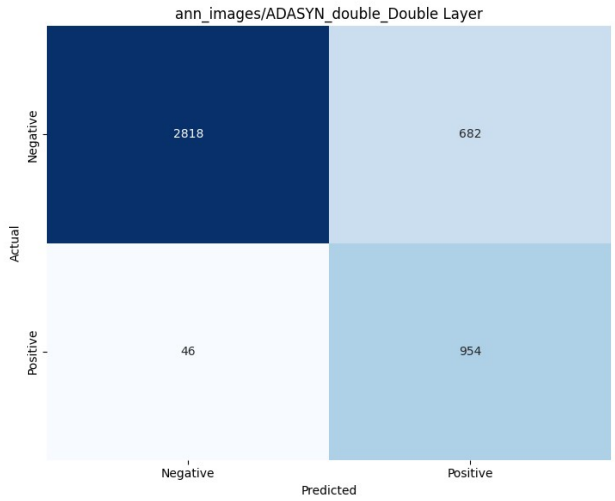
Accuracy: 0.89
Precision: 0.86
Recall (Sensitivity): 0.92
Specificity: 0.85
F1 Score: 0.89



Metrics

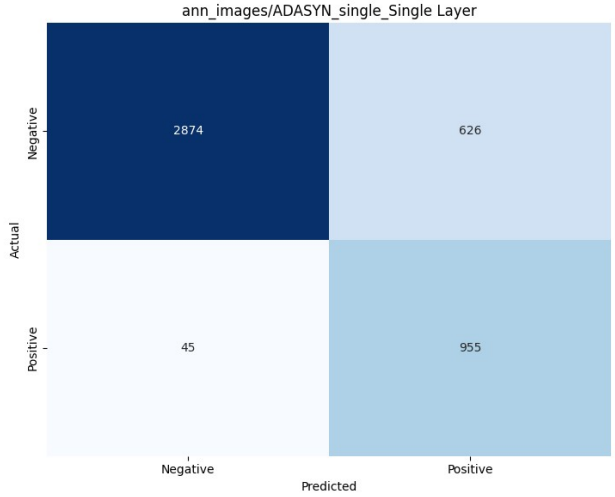
Accuracy: 0.89
Precision: 0.86
Recall (Sensitivity): 0.93
Specificity: 0.85
F1 Score: 0.89

ADASYN Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



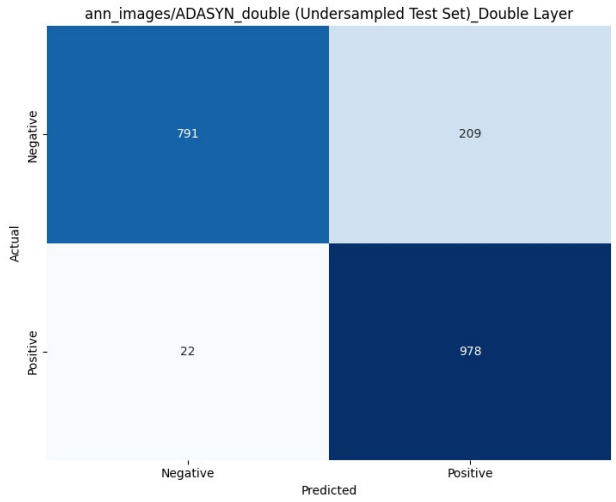
Metrics

Accuracy: 0.84
Precision: 0.58
Recall (Sensitivity): 0.95
Specificity: 0.81
F1 Score: 0.72



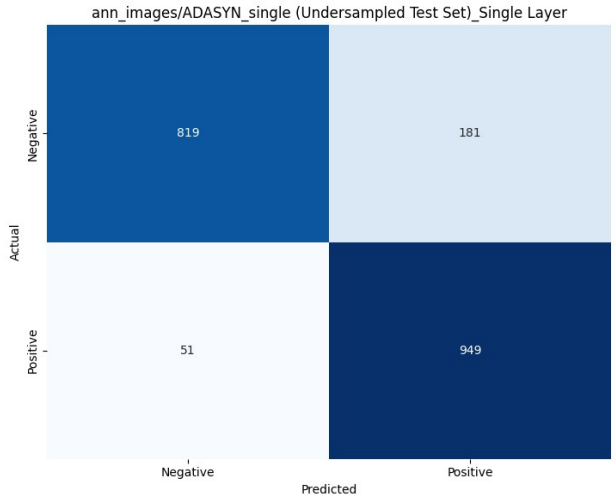
Metrics

Accuracy: 0.85
Precision: 0.60
Recall (Sensitivity): 0.95
Specificity: 0.82
F1 Score: 0.74



Metrics

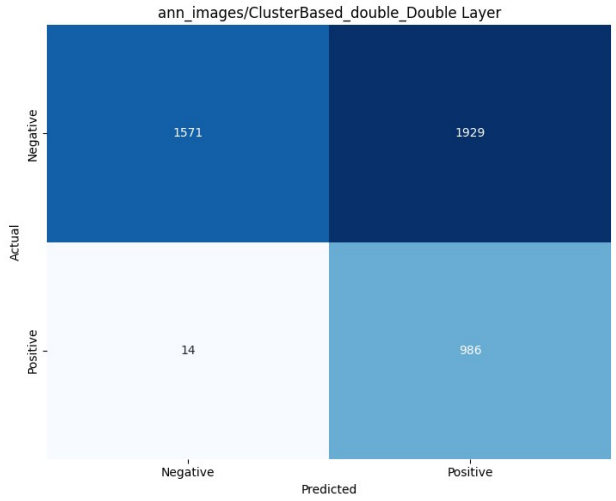
Accuracy: 0.88
Precision: 0.82
Recall (Sensitivity): 0.98
Specificity: 0.79
F1 Score: 0.89



Metrics

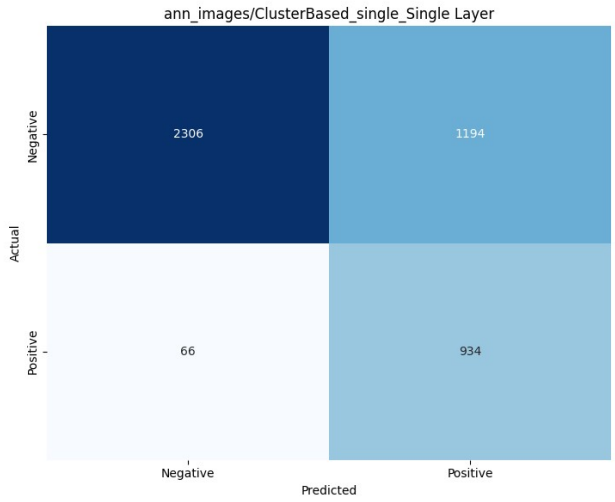
Accuracy: 0.88
Precision: 0.84
Recall (Sensitivity): 0.95
Specificity: 0.82
F1 Score: 0.89

Cluster Based Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



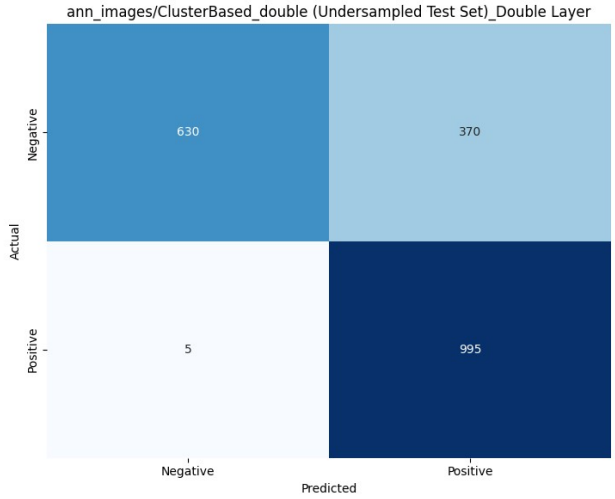
Metrics

Accuracy: 0.57
Precision: 0.34
Recall (Sensitivity): 0.99
Specificity: 0.45
F1 Score: 0.50



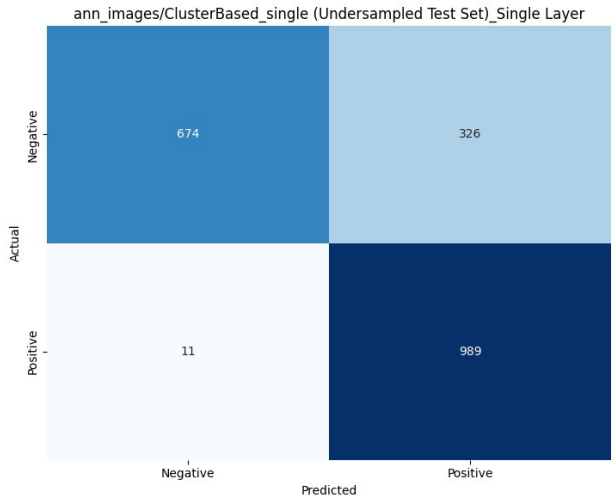
Metrics

Accuracy: 0.72
Precision: 0.44
Recall (Sensitivity): 0.93
Specificity: 0.66
F1 Score: 0.60



Metrics

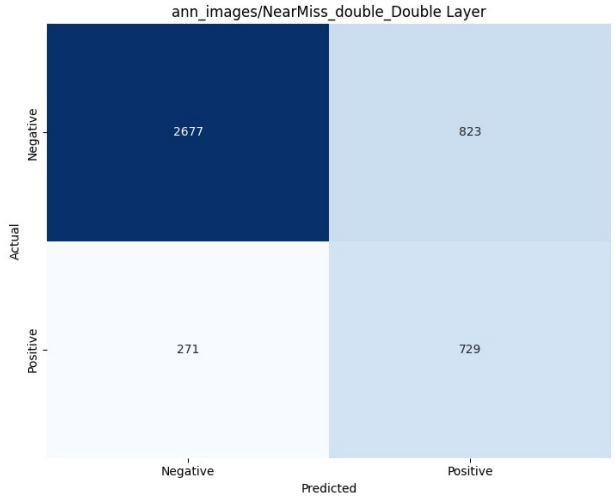
Accuracy: 0.81
Precision: 0.73
Recall (Sensitivity): 0.99
Specificity: 0.63
F1 Score: 0.84



Metrics

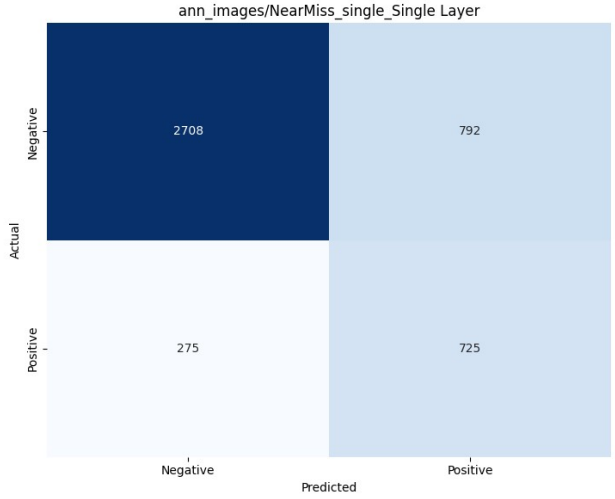
Accuracy: 0.83
Precision: 0.75
Recall (Sensitivity): 0.99
Specificity: 0.67
F1 Score: 0.85

NearMiss Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



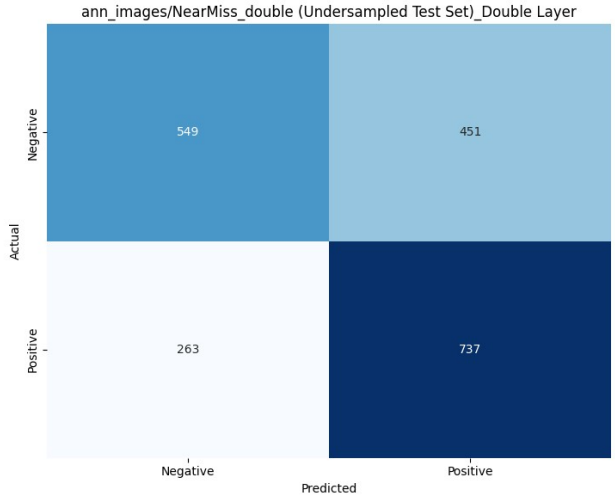
Metrics

Accuracy: 0.76
Precision: 0.47
Recall (Sensitivity): 0.73
Specificity: 0.76
F1 Score: 0.57



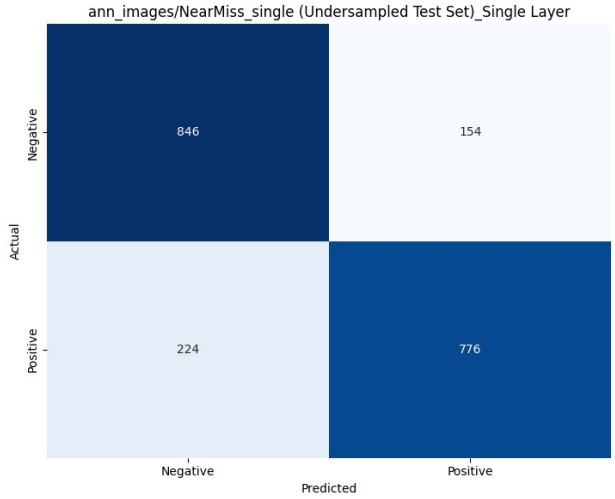
Metrics

Accuracy: 0.76
Precision: 0.48
Recall (Sensitivity): 0.72
Specificity: 0.77
F1 Score: 0.58



Metrics

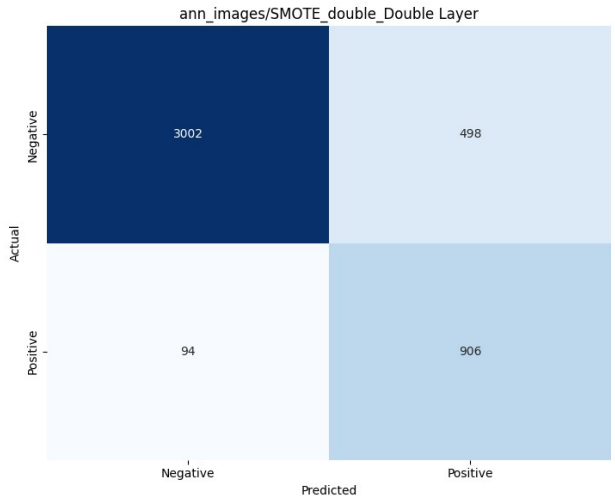
Accuracy: 0.64
Precision: 0.62
Recall (Sensitivity): 0.74
Specificity: 0.55
F1 Score: 0.67



Metrics

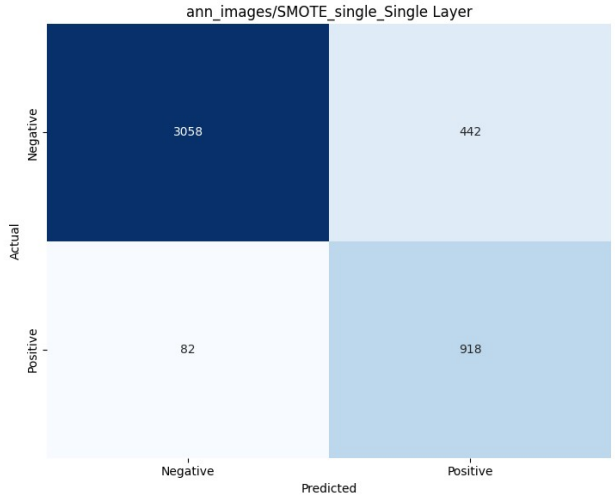
Accuracy: 0.81
Precision: 0.83
Recall (Sensitivity): 0.78
Specificity: 0.85
F1 Score: 0.80

SMOTE Single Layer and Double Layer Models With Undersampled Test Set and original Test Set:



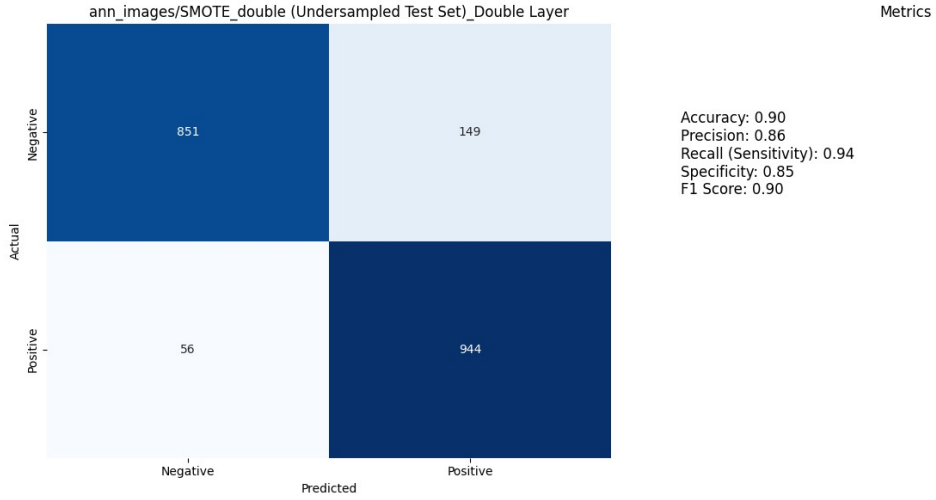
Metrics

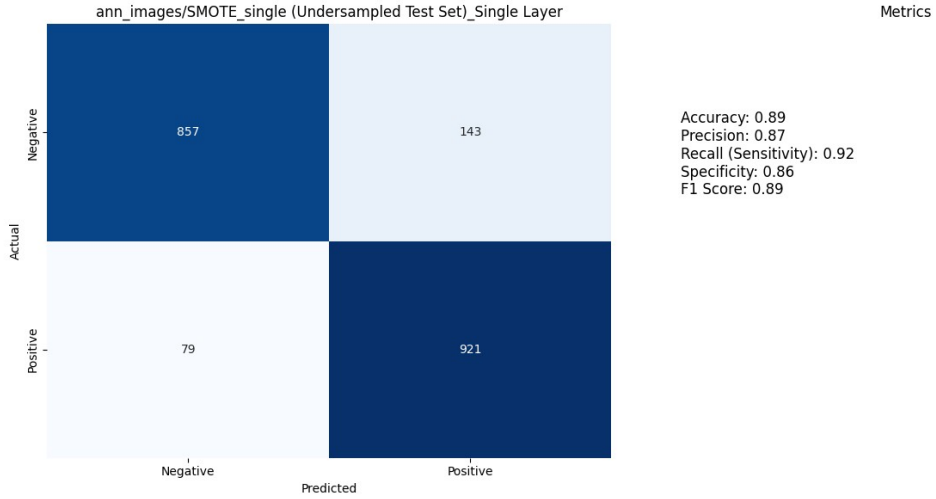
Accuracy: 0.87
Precision: 0.65
Recall (Sensitivity): 0.91
Specificity: 0.86
F1 Score: 0.75



Metrics

Accuracy: 0.88
Precision: 0.68
Recall (Sensitivity): 0.92
Specificity: 0.87
F1 Score: 0.78

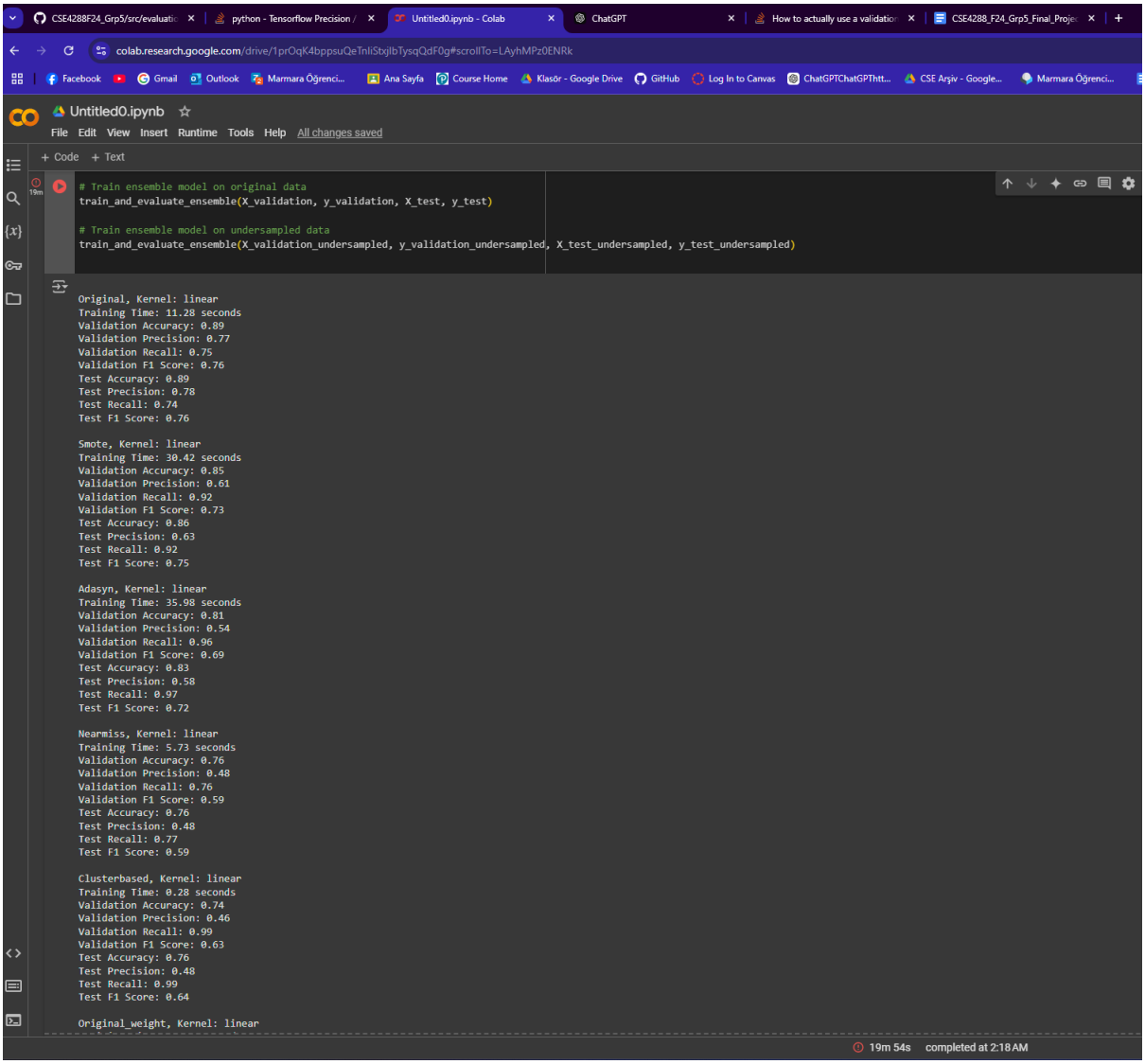




SVM (Support Vector Machine)

The SMV model is used with a linear kernel. Weighted classifications were introduced to address class imbalance. Models trained on resampled datasets were compared against the original dataset. An ensemble method, VotingClassifier, combined predictions from multiple SVM models.

Due to the program's execution time being enormous, I prefer to choose only 1 hyperparameter which is the kernel.



Kernel = Linear (train data is original,smote,adasyn,nearmiss,clusterbased and weight assigned to original data):

```
Original, Kernel: linear
Training Time: 11.28 seconds
Validation Accuracy: 0.89
Validation Precision: 0.77
Validation Recall: 0.75
Validation F1 Score: 0.76
Test Accuracy: 0.89
Test Precision: 0.78
Test Recall: 0.74
Test F1 Score: 0.76

Smote, Kernel: linear
Training Time: 30.42 seconds
Validation Accuracy: 0.85
Validation Precision: 0.61
Validation Recall: 0.92
Validation F1 Score: 0.73
Test Accuracy: 0.86
Test Precision: 0.63
Test Recall: 0.92
Test F1 Score: 0.75

Adasyn, Kernel: linear
Training Time: 35.98 seconds
Validation Accuracy: 0.81
Validation Precision: 0.54
Validation Recall: 0.96
Validation F1 Score: 0.69
Test Accuracy: 0.83
Test Precision: 0.58
Test Recall: 0.97
Test F1 Score: 0.72

Nearmiss, Kernel: linear
Training Time: 5.73 seconds
Validation Accuracy: 0.76
Validation Precision: 0.48
Validation Recall: 0.76
Validation F1 Score: 0.59
Test Accuracy: 0.76
Test Precision: 0.48
Test Recall: 0.77
Test F1 Score: 0.59

Clusterbased, Kernel: linear
Training Time: 0.28 seconds
Validation Accuracy: 0.74
Validation Precision: 0.46
Validation Recall: 0.99
Validation F1 Score: 0.63
Test Accuracy: 0.76
Test Precision: 0.48
Test Recall: 0.99
Test F1 Score: 0.64

Original_weight, Kernel: linear
Training Time: 15.28 seconds
Validation Accuracy: 0.85
Validation Precision: 0.60
Validation Recall: 0.93
Validation F1 Score: 0.73
Test Accuracy: 0.86
Test Precision: 0.63
Test Recall: 0.92
Test F1 Score: 0.75
```

Kernel = Linear (Undersampled test and validation)

```
Original_undersampled, Kernel: linear
Training Time: 10.31 seconds
Validation Accuracy: 0.84
Validation Precision: 0.92
Validation Recall: 0.75
Validation F1 Score: 0.83
Test Accuracy: 0.84
Test Precision: 0.92
Test Recall: 0.74
Test F1 Score: 0.82

Smote_undersampled, Kernel: linear
Training Time: 29.29 seconds
Validation Accuracy: 0.88
Validation Precision: 0.85
Validation Recall: 0.92
Validation F1 Score: 0.88
Test Accuracy: 0.88
Test Precision: 0.86
Test Recall: 0.92
Test F1 Score: 0.89

Adasyn_undersampled, Kernel: linear
Training Time: 34.49 seconds
Validation Accuracy: 0.87
Validation Precision: 0.81
Validation Recall: 0.96
Validation F1 Score: 0.88
Test Accuracy: 0.88
Test Precision: 0.82
Test Recall: 0.97
Test F1 Score: 0.89

Nearmiss_undersampled, Kernel: linear
Training Time: 5.73 seconds
Validation Accuracy: 0.77
Validation Precision: 0.77
Validation Recall: 0.76
Validation F1 Score: 0.77
Test Accuracy: 0.76
Test Precision: 0.76
Test Recall: 0.77
Test F1 Score: 0.76

Clusterbased_undersampled, Kernel: linear
Training Time: 0.41 seconds
Validation Accuracy: 0.83
Validation Precision: 0.75
Validation Recall: 0.99
Validation F1 Score: 0.85
Test Accuracy: 0.84
Test Precision: 0.76
Test Recall: 0.99
Test F1 Score: 0.86
```

Kernel = RBF:

```
Original, Kernel: rbf
Training Time: 17.08 seconds
Validation Accuracy: 0.91
Validation Precision: 0.83
Validation Recall: 0.74
Validation F1 Score: 0.78
Test Accuracy: 0.91
Test Precision: 0.83
Test Recall: 0.75
Test F1 Score: 0.79

Nearmiss, Kernel: rbf
Training Time: 10.63 seconds
Validation Accuracy: 0.41
Validation Precision: 0.24
Validation Recall: 0.77
Validation F1 Score: 0.37
Test Accuracy: 0.39
Test Precision: 0.24
Test Recall: 0.78
Test F1 Score: 0.36

Smote, Kernel: rbf
Training Time: 43.27 seconds
Validation Accuracy: 0.86
Validation Precision: 0.63
Validation Recall: 0.92
Validation F1 Score: 0.75
Test Accuracy: 0.88
Test Precision: 0.65
Test Recall: 0.93
Test F1 Score: 0.77

Clusterbased, Kernel: rbf
Training Time: 0.47 seconds
Validation Accuracy: 0.73
Validation Precision: 0.45
Validation Recall: 0.99
Validation F1 Score: 0.62
Test Accuracy: 0.75
Test Precision: 0.47
Test Recall: 0.99
Test F1 Score: 0.64

Adasyn, Kernel: rbf
Training Time: 54.45 seconds
Validation Accuracy: 0.81
Validation Precision: 0.54
Validation Recall: 0.97
Validation F1 Score: 0.69
Test Accuracy: 0.82
Test Precision: 0.56
Test Recall: 0.98
Test F1 Score: 0.71

Original_weight, Kernel: rbf
Training Time: 22.80 seconds
Validation Accuracy: 0.86
Validation Precision: 0.62
Validation Recall: 0.93
Validation F1 Score: 0.74
Test Accuracy: 0.87
Test Precision: 0.64
Test Recall: 0.94
Test F1 Score: 0.77
```

Kernel = RBF (Undersampled test and validation)

```
Original_undersampled, Kernel: rbf
Training Time: 13.47 seconds
Validation Accuracy: 0.85
Validation Precision: 0.94
Validation Recall: 0.74
Validation F1 Score: 0.83
Test Accuracy: 0.85
Test Precision: 0.95
Test Recall: 0.75
Test F1 Score: 0.84

Smote_undersampled, Kernel: rbf
Training Time: 38.98 seconds
Validation Accuracy: 0.88
Validation Precision: 0.86
Validation Recall: 0.92
Validation F1 Score: 0.89
Test Accuracy: 0.90
Test Precision: 0.87
Test Recall: 0.93
Test F1 Score: 0.90

Adasyn_undersampled, Kernel: rbf
Training Time: 48.90 seconds
Validation Accuracy: 0.86
Validation Precision: 0.80
Validation Recall: 0.97
Validation F1 Score: 0.88
Test Accuracy: 0.88
Test Precision: 0.81
Test Recall: 0.98
Test F1 Score: 0.89

Nearmiss_undersampled, Kernel: rbf
Training Time: 7.01 seconds
Validation Accuracy: 0.54
Validation Precision: 0.53
Validation Recall: 0.77
Validation F1 Score: 0.62
Test Accuracy: 0.53
Test Precision: 0.52
Test Recall: 0.78
Test F1 Score: 0.62

Clusterbased_undersampled, Kernel: rbf
Training Time: 0.53 seconds
Validation Accuracy: 0.82
Validation Precision: 0.74
Validation Recall: 0.99
Validation F1 Score: 0.85
Test Accuracy: 0.84
Test Precision: 0.76
Test Recall: 0.99
Test F1 Score: 0.86
```

Kernel = Poly:

```
Original, Kernel: poly
Training Time: 10.55 seconds
Validation Accuracy: 0.91
Validation Precision: 0.84
Validation Recall: 0.73
Validation F1 Score: 0.78
Test Accuracy: 0.91
Test Precision: 0.84
Test Recall: 0.75
Test F1 Score: 0.79

Nearmiss, Kernel: poly
Training Time: 5.94 seconds
Validation Accuracy: 0.81
Validation Precision: 0.56
Validation Recall: 0.76
Validation F1 Score: 0.64
Test Accuracy: 0.82
Test Precision: 0.58
Test Recall: 0.77
Test F1 Score: 0.66

Smote, Kernel: poly
Training Time: 30.36 seconds
Validation Accuracy: 0.86
Validation Precision: 0.63
Validation Recall: 0.92
Validation F1 Score: 0.75
Test Accuracy: 0.88
Test Precision: 0.66
Test Recall: 0.93
Test F1 Score: 0.77

Clusterbased, Kernel: poly
Training Time: 0.25 seconds
Validation Accuracy: 0.73
Validation Precision: 0.45
Validation Recall: 0.99
Validation F1 Score: 0.62
Test Accuracy: 0.75
Test Precision: 0.47
Test Recall: 0.99
Test F1 Score: 0.64

Adasyn, Kernel: poly
Training Time: 37.62 seconds
Validation Accuracy: 0.81
Validation Precision: 0.54
Validation Recall: 0.97
Validation F1 Score: 0.69
Test Accuracy: 0.83
Test Precision: 0.56
Test Recall: 0.98
Test F1 Score: 0.71

Original_weight, Kernel: poly
Training Time: 14.85 seconds
Validation Accuracy: 0.86
Validation Precision: 0.62
Validation Recall: 0.93
Validation F1 Score: 0.74
Test Accuracy: 0.87
Test Precision: 0.65
Test Recall: 0.94
Test F1 Score: 0.77
```

Kernel = Poly (Undersampled test and validation):

```
Original_undersampled, Kernel: poly
Training Time: 9.66 seconds
Validation Accuracy: 0.84
Validation Precision: 0.94
Validation Recall: 0.73
Validation F1 Score: 0.82
Test Accuracy: 0.86
Test Precision: 0.95
Test Recall: 0.75
Test F1 Score: 0.84

Smote_undersampled, Kernel: poly
Training Time: 27.00 seconds
Validation Accuracy: 0.88
Validation Precision: 0.86
Validation Recall: 0.92
Validation F1 Score: 0.89
Test Accuracy: 0.89
Test Precision: 0.87
Test Recall: 0.93
Test F1 Score: 0.90

Adasyn_undersampled, Kernel: poly
Training Time: 36.66 seconds
Validation Accuracy: 0.86
Validation Precision: 0.80
Validation Recall: 0.97
Validation F1 Score: 0.88
Test Accuracy: 0.88
Test Precision: 0.81
Test Recall: 0.98
Test F1 Score: 0.89

Nearmiss_undersampled, Kernel: poly
Training Time: 4.52 seconds
Validation Accuracy: 0.79
Validation Precision: 0.81
Validation Recall: 0.76
Validation F1 Score: 0.78
Test Accuracy: 0.80
Test Precision: 0.82
Test Recall: 0.77
Test F1 Score: 0.80

Clusterbased_undersampled, Kernel: poly
Training Time: 0.20 seconds
Validation Accuracy: 0.82
Validation Precision: 0.74
Validation Recall: 0.99
Validation F1 Score: 0.85
Test Accuracy: 0.84
Test Precision: 0.76
Test Recall: 0.99
Test F1 Score: 0.86
```

Kernel = Sigmoid:

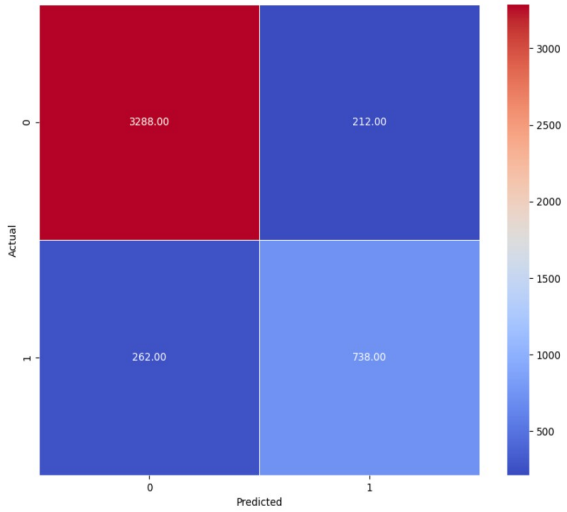
```
Original, Kernel: sigmoid
Training Time: 28.09 seconds
Validation Accuracy: 0.78
Validation Precision: 0.50
Validation Recall: 0.51
Validation F1 Score: 0.51
Test Accuracy: 0.78
Test Precision: 0.51
Test Recall: 0.49
Test F1 Score: 0.50

Nearmiss, Kernel: sigmoid
Training Time: 14.20 seconds
Validation Accuracy: 0.61
Validation Precision: 0.31
Validation Recall: 0.62
Validation F1 Score: 0.41
Test Accuracy: 0.61
Test Precision: 0.30
Test Recall: 0.60
Test F1 Score: 0.40

Smote, Kernel: sigmoid
Training Time: 108.89 seconds
Validation Accuracy: 0.73
Validation Precision: 0.44
Validation Recall: 0.75
Validation F1 Score: 0.55
Test Accuracy: 0.74
Test Precision: 0.45
Test Recall: 0.74
Test F1 Score: 0.56

Clusterbased, Kernel: sigmoid
Training Time: 1.80 seconds
Validation Accuracy: 0.73
Validation Precision: 0.45
Validation Recall: 0.94
Validation F1 Score: 0.60
Test Accuracy: 0.74
Test Precision: 0.46
Test Recall: 0.93
Test F1 Score: 0.61

Adasyn, Kernel: sigmoid
Training Time: 125.99 seconds
Validation Accuracy: 0.71
Validation Precision: 0.42
Validation Recall: 0.74
Validation F1 Score: 0.53
Test Accuracy: 0.72
Test Precision: 0.43
Test Recall: 0.74
Test F1 Score: 0.54

Original_weight, Kernel: sigmoid
Training Time: 38.83 seconds
Validation Accuracy: 0.73
Validation Precision: 0.44
Validation Recall: 0.75
Validation F1 Score: 0.56
Test Accuracy: 0.74
Test Precision: 0.45
Test Recall: 0.75
Test F1 Score: 0.57
```

Kernel = Sigmoid (Undersampled test and validation):

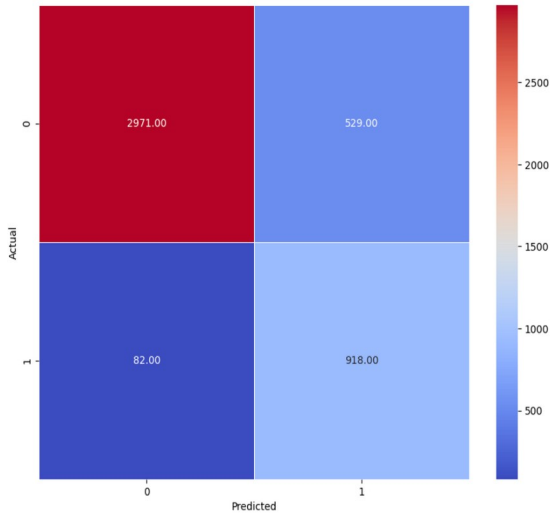
```
Original_undersampled, Kernel: sigmoid
Training Time: 23.03 seconds
Validation Accuracy: 0.68
Validation Precision: 0.78
Validation Recall: 0.51
Validation F1 Score: 0.62
Test Accuracy: 0.68
Test Precision: 0.79
Test Recall: 0.49
Test F1 Score: 0.61

Smote_undersampled, Kernel: sigmoid
Training Time: 103.00 seconds
Validation Accuracy: 0.74
Validation Precision: 0.74
Validation Recall: 0.75
Validation F1 Score: 0.74
Test Accuracy: 0.74
Test Precision: 0.74
Test Recall: 0.74
Test F1 Score: 0.74

Adasyn_undersampled, Kernel: sigmoid
Training Time: 122.23 seconds
Validation Accuracy: 0.73
Validation Precision: 0.72
Validation Recall: 0.74
Validation F1 Score: 0.73
Test Accuracy: 0.72
Test Precision: 0.71
Test Recall: 0.74
Test F1 Score: 0.73

Nearmiss_undersampled, Kernel: sigmoid
Training Time: 12.04 seconds
Validation Accuracy: 0.61
Validation Precision: 0.61
Validation Recall: 0.62
Validation F1 Score: 0.61
Test Accuracy: 0.61
Test Precision: 0.61
Test Recall: 0.60
Test F1 Score: 0.60

Clusterbased_undersampled, Kernel: sigmoid
Training Time: 1.40 seconds
Validation Accuracy: 0.80
Validation Precision: 0.73
Validation Recall: 0.94
Validation F1 Score: 0.82
Test Accuracy: 0.81
Test Precision: 0.75
Test Recall: 0.93
Test F1 Score: 0.83
```

Confusion Matrices for Kernel = Linear

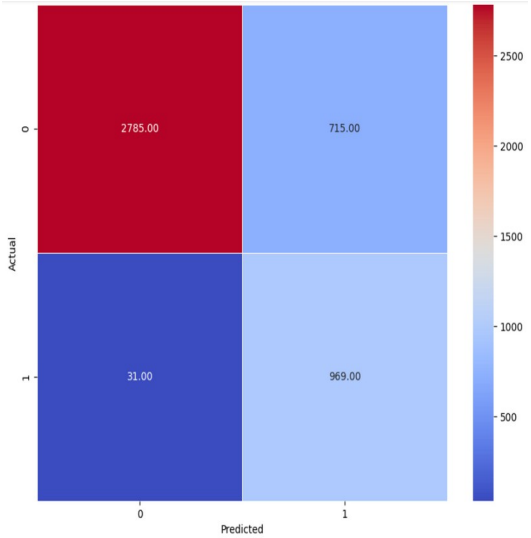
Original (train_data_ID.csv):



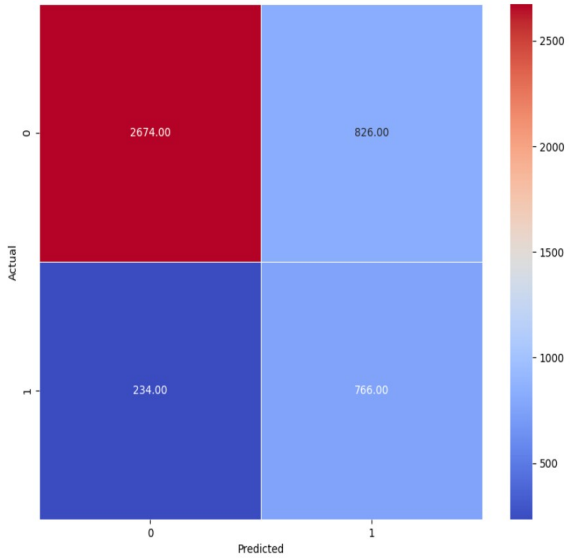
Smote (train_data_ID_smote.csv):



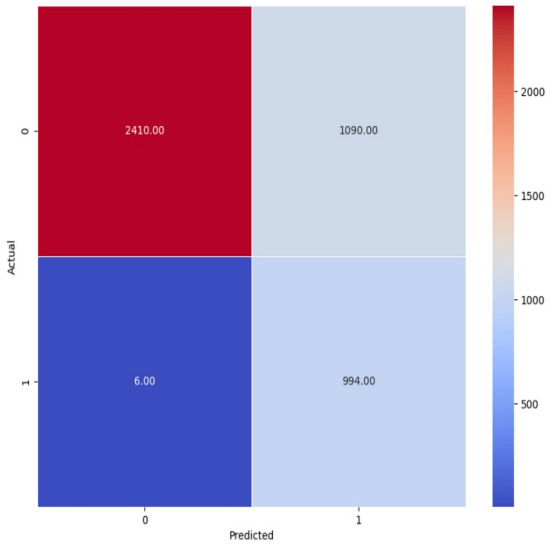
Adasyn (train_data_ID_adasyn.csv):



Nearmiss (train_data_ID_nearmiss.csv):

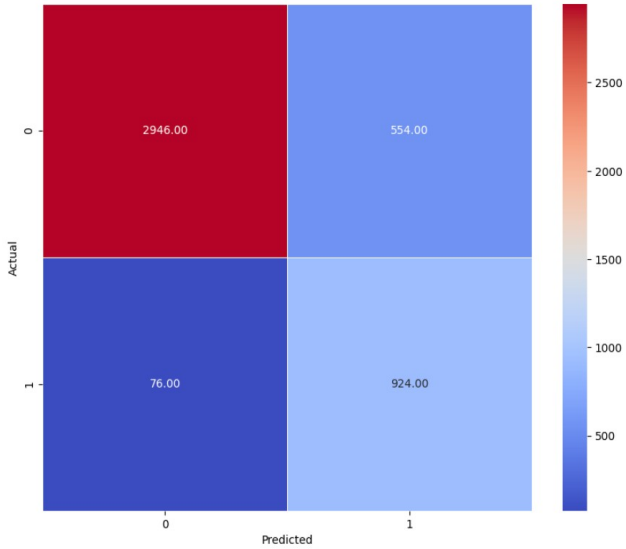


Clusterbased (train_data_ID_clusterbased.csv):



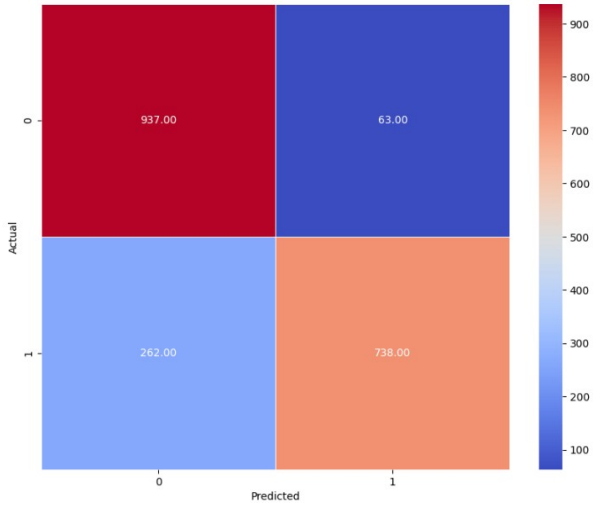
Original Data with weight assigned (copy of original data):

In this case, weights assigned to the original data inside of the train part with a hyperparameter of SVC.

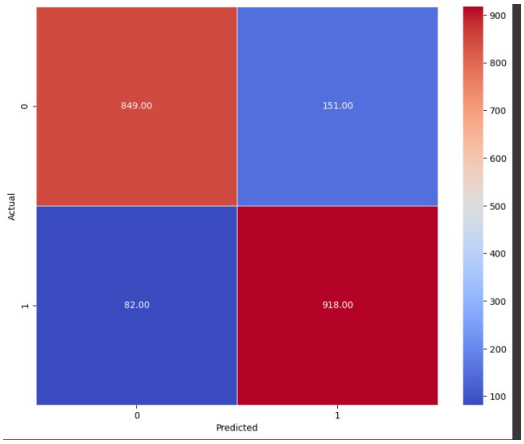


Undersampled Tests and Validations:

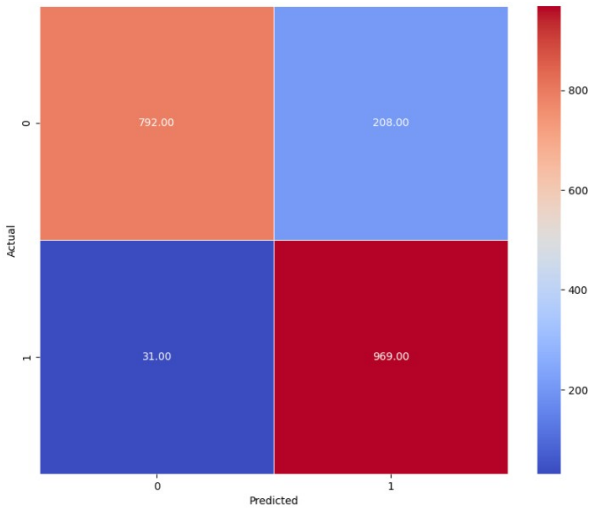
Original:



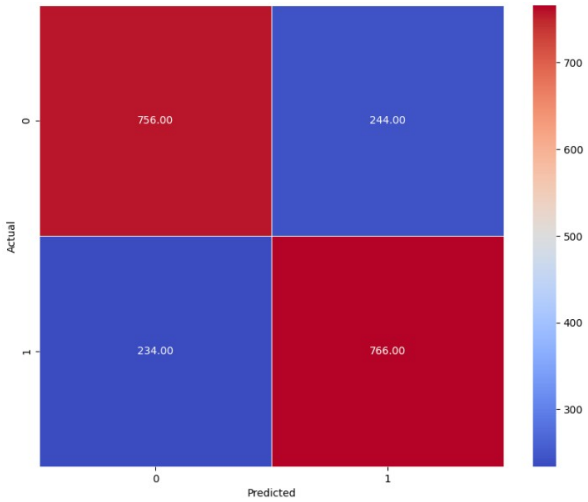
Smote:



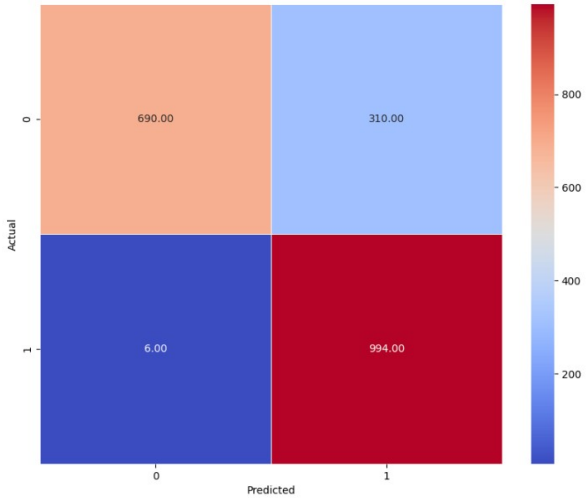
Adasyn:



Nearmiss:



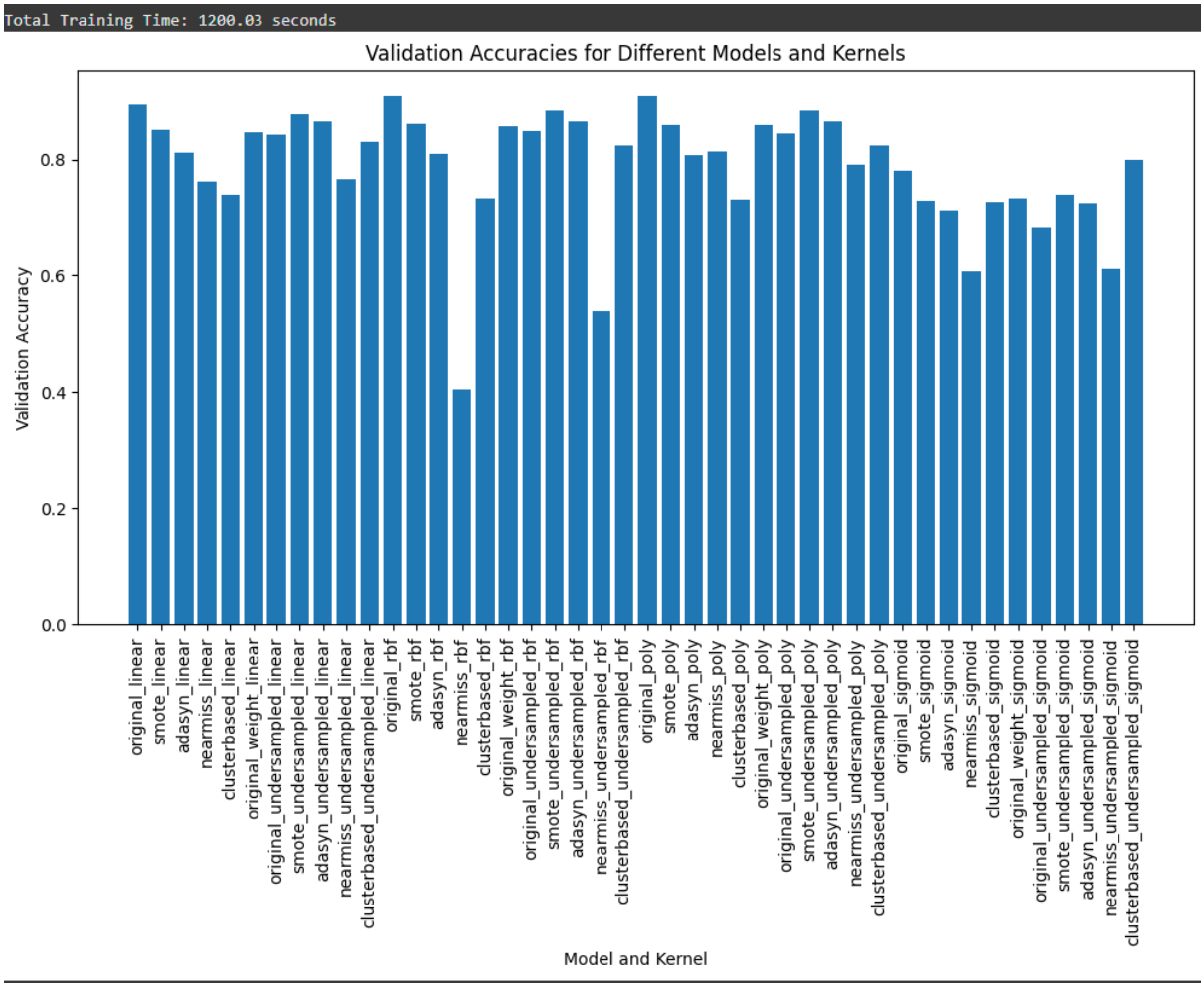
Clusterbased:



According to the all train datasets
(6 (original test and validation) / 5 (undersampled test and validation)),
test and validation datasets (2);

validation accuracies calculated based on kernel hyperparameter
(has 4 parameter; Linear, RBF, Poly, Sigmoid)

In total: (6 x 4 + 5 x 4) = 44 Model trained.



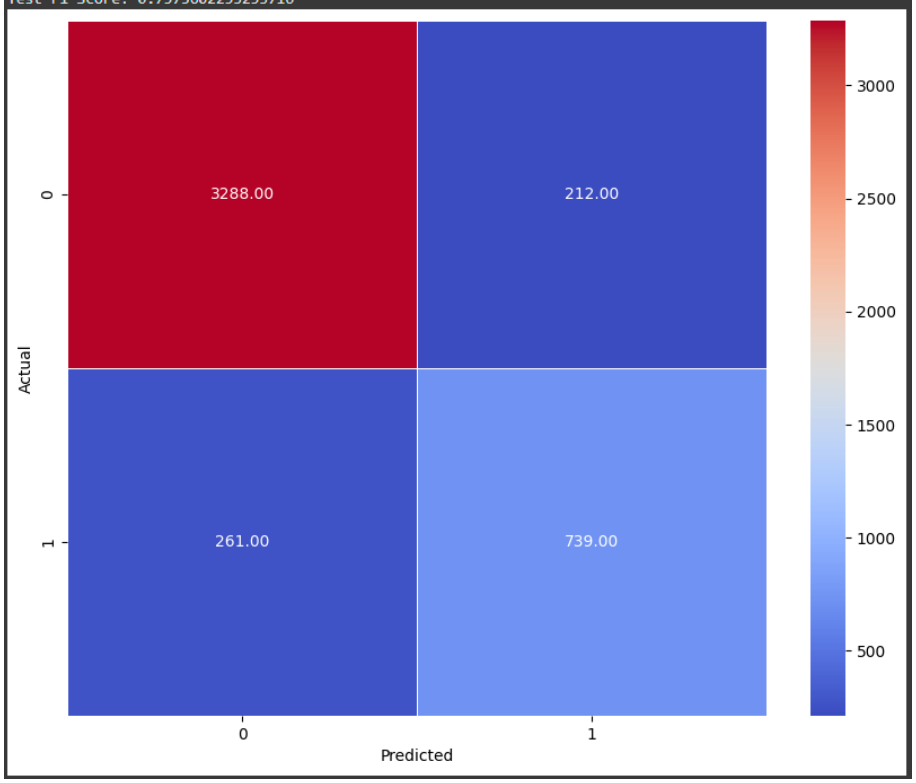
For Kernel = Linear:

-The model with highest **F1-Score** is trained with **adasyn** dataset and **undersampled** test and validation datasets.

-The model with highest **Validation Accuracy** is trained with **original** dataset and **non-undersampled** test and validation datasets.

VotingClassifier (Soft): Original Dataset

```
Ensemble Model (Kernel: linear):
Validation Accuracy: 0.8937777777777778
Test Accuracy: 0.8948888888888888

Validation Precision: 0.7665985699693565
Validation Recall: 0.7505
Validation F1 Score: 0.7584638706417383

Test Precision: 0.7770767613038907
Test Recall: 0.739
Test F1 Score: 0.7575602255253716
```



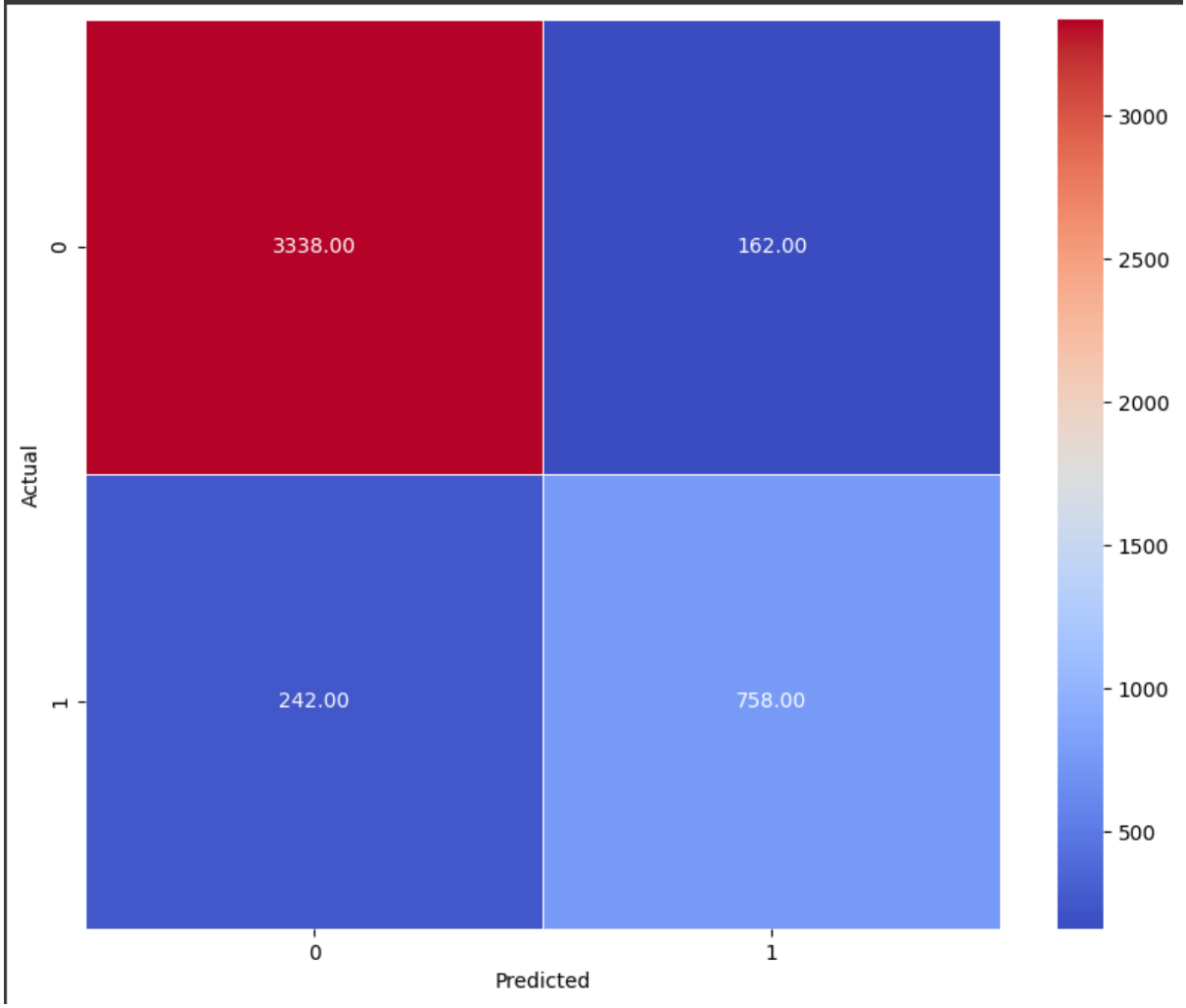
For Kernel = RBF:

-The model with highest **F1-Score** is trained with **Smote** dataset and **undersampled** test and validation datasets.

-The model with highest **Validation Accuracy** is trained with **original** dataset and **non-undersampled** test and validation datasets.

VotingClassifier (Soft): Original

```
Ensemble Model (Kernel: rbf):
Validation Accuracy: 0.9085555555555556
Test Accuracy: 0.9102222222222223

Validation Precision: 0.8224657534246576
Validation Recall: 0.7505
Validation F1 Score: 0.7848366013071896

Test Precision: 0.8239130434782609
Test Recall: 0.758
Test F1 Score: 0.7895833333333333
```

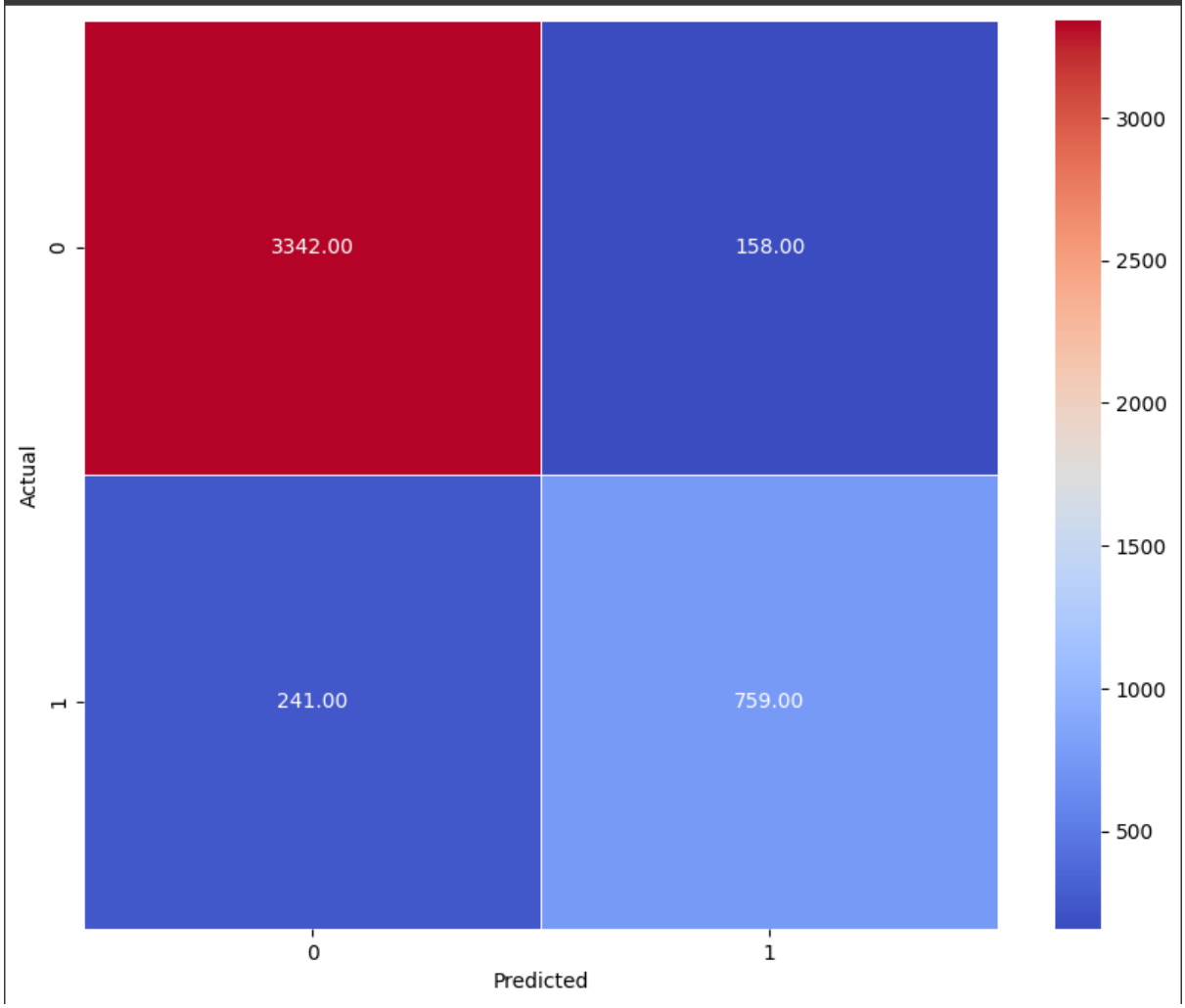


For Kernel = Poly:

-The model with highest **F1-Score** is trained with **adasyn** dataset and **undersampled** test and validation datasets.
-The model with highest **Validation Accuracy** is trained with **original** dataset and **non-undersampled** test and validation datasets.

VotingClassifier(Soft): Original

```
Ensemble Model (Kernel: poly):
Validation Accuracy: 0.9074444444444445
Test Accuracy: 0.9113333333333333

Validation Precision: 0.8272574312955693
Validation Recall: 0.7375
Validation F1 Score: 0.7798043880518107

Test Precision: 0.8276990185387132
Test Recall: 0.759
Test F1 Score: 0.7918622848200313
```



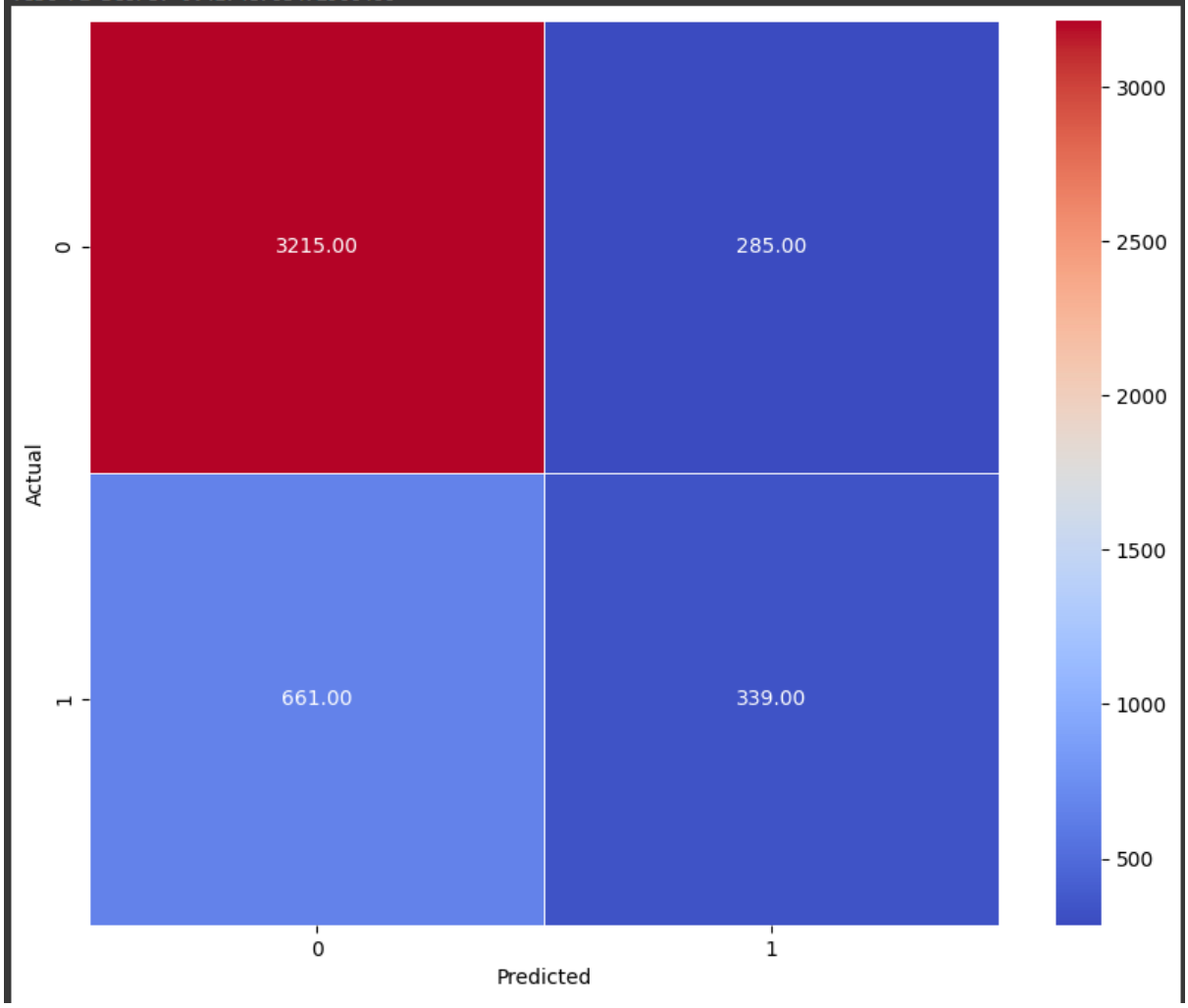
For Kernel = Sigmoid:

-The model with highest **F1-Score** is trained with **adasyn** dataset and **undersampled** test and validation datasets.

-The model with highest **Validation Accuracy** is trained with **original** dataset and **non-undersampled** test and validation datasets.

VotingClassifier(Soft): Original

```
Ensemble Model (Kernel: sigmoid):
Validation Accuracy: 0.7926666666666666
Test Accuracy: 0.7897777777777778

Validation Precision: 0.5507575757575758
Validation Recall: 0.3635
Validation F1 Score: 0.43795180722891563

Test Precision: 0.5432692307692307
Test Recall: 0.339
Test F1 Score: 0.41748768472906406
```



According to the their F1 Scores, best hyperparametes;
For Original Dataset: Both RBF and Poly (16.44 secs, 10.90 secs)

For Smote applied dataset: Both RBF and Poly (45.39 secs ,29.24 secs)

For Adasyn applied dataset: Linear (36.45 secs)

For Nearmiss applied dataset: Poly (6.66)

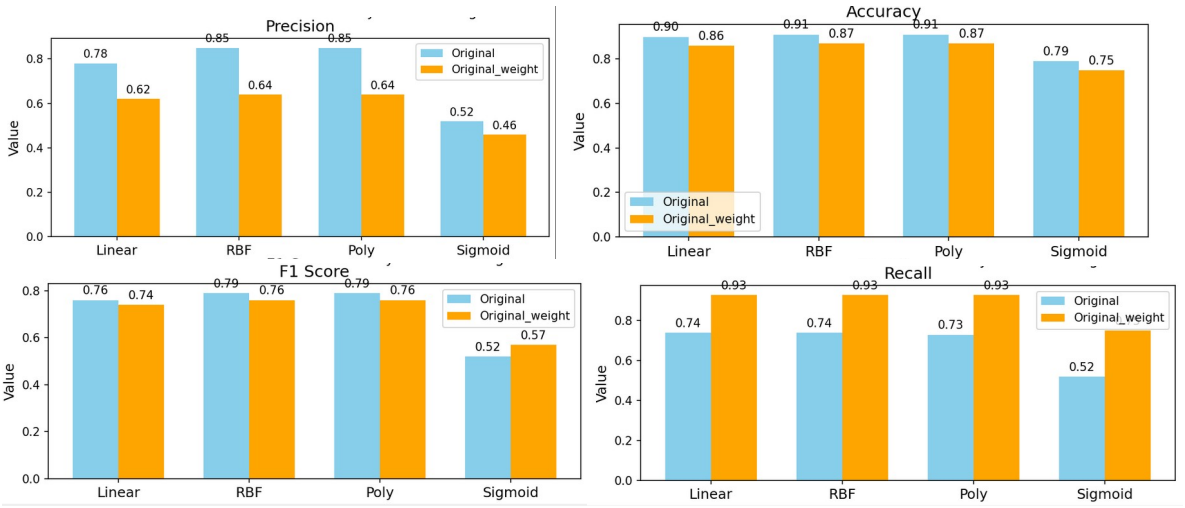
For Clusterbased applied dataset: Both RBF and Poly (0.48 secs, 0.25 secs)

In summary, when the hyperparameter kernel equals the “poly”, performance and execution time is the best value of the kernel for most of the models for loan approval dataset.

Cross Validation for SVM

Cross-validation is a resampling method that tests the model using unseen sub-datasets. 10-fold cross-validation was applied to measure test performance. This means the dataset is divided into 10 sub-datasets. The other 9 subdatasets are tested with one subdataset. *The Stratified K-Fold* method ensured a balanced distribution of data classes. It ensures that the fold consists of approximately the same percentage of the class label in each fold.

In the project, the SVM method was tested using four different kernel functions (linear, rbf, poly, sigmoid). Also, model performance was evaluated in two other cases with and without class weighting

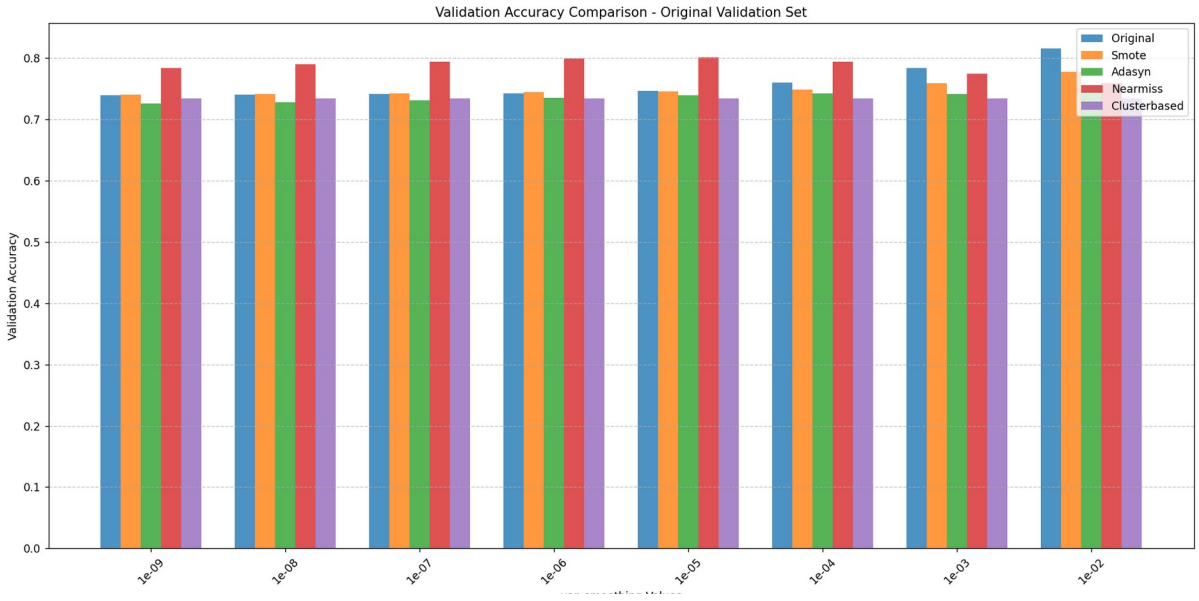


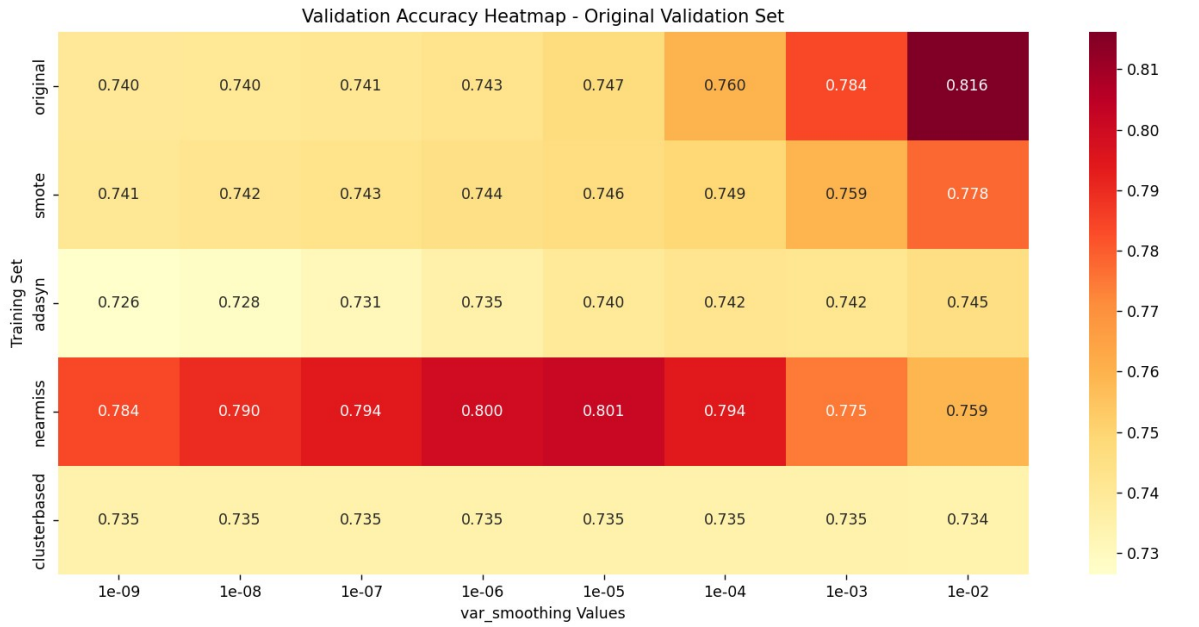
Gaussian Naive Bayes

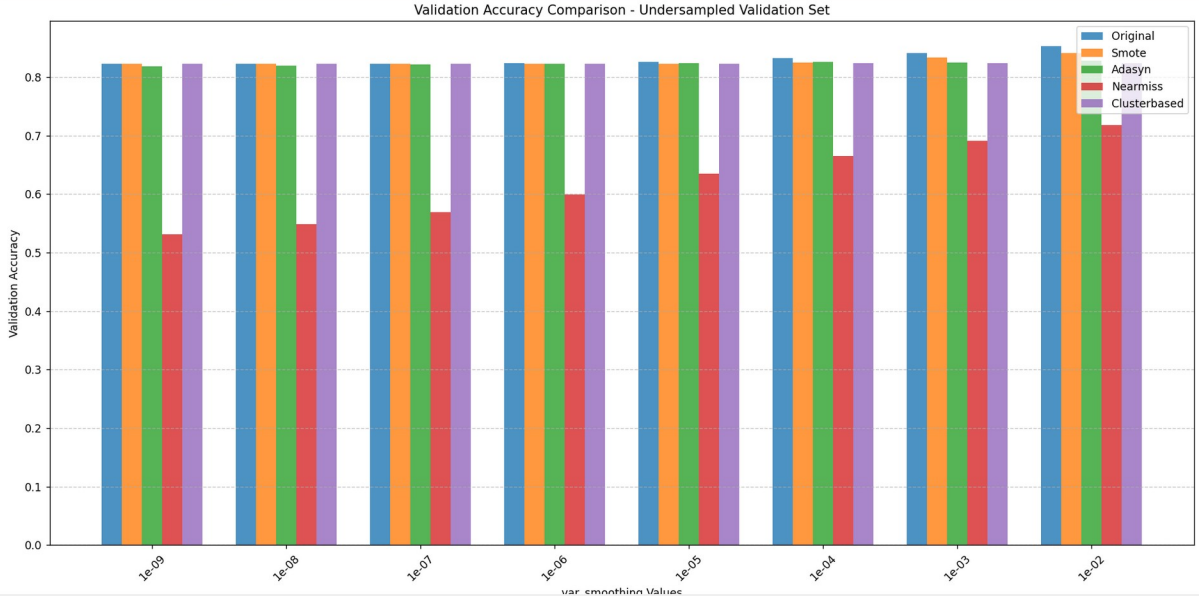
This study implements the Gaussian Naive Bayes (GNB) algorithm to evaluate its performance across five training datasets. The GNB model assumes that the features follow a Gaussian (normal) distribution and calculates class probabilities based on this assumption. For each dataset, the model was trained on the training data and tested on both validation and test datasets.

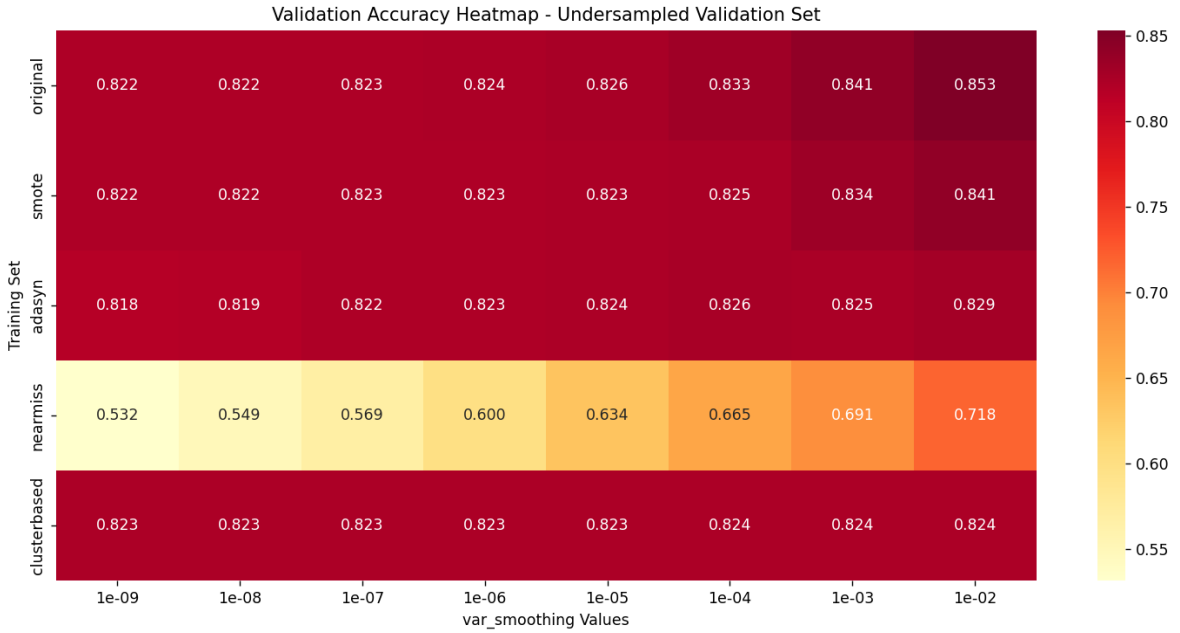
The model's performance was assessed using accuracy as the evaluation metric. Validation datasets were utilized to analyze how the Gaussian assumption affected classification performance across different datasets. The results highlight the

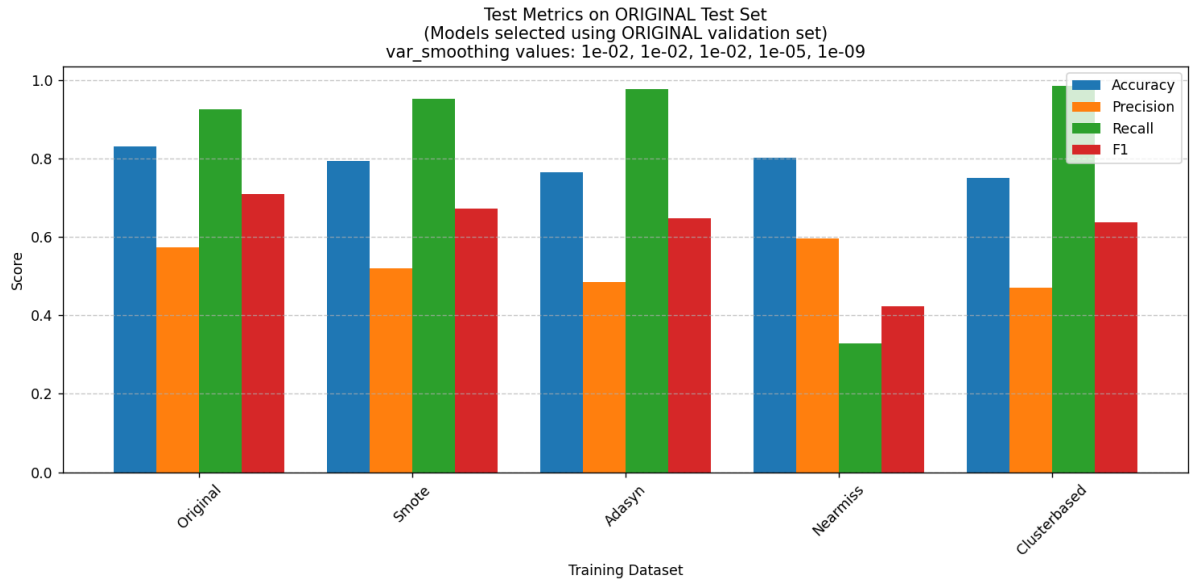
strengths and limitations of GNB, particularly in datasets where feature independence and Gaussian distribution assumptions hold true.

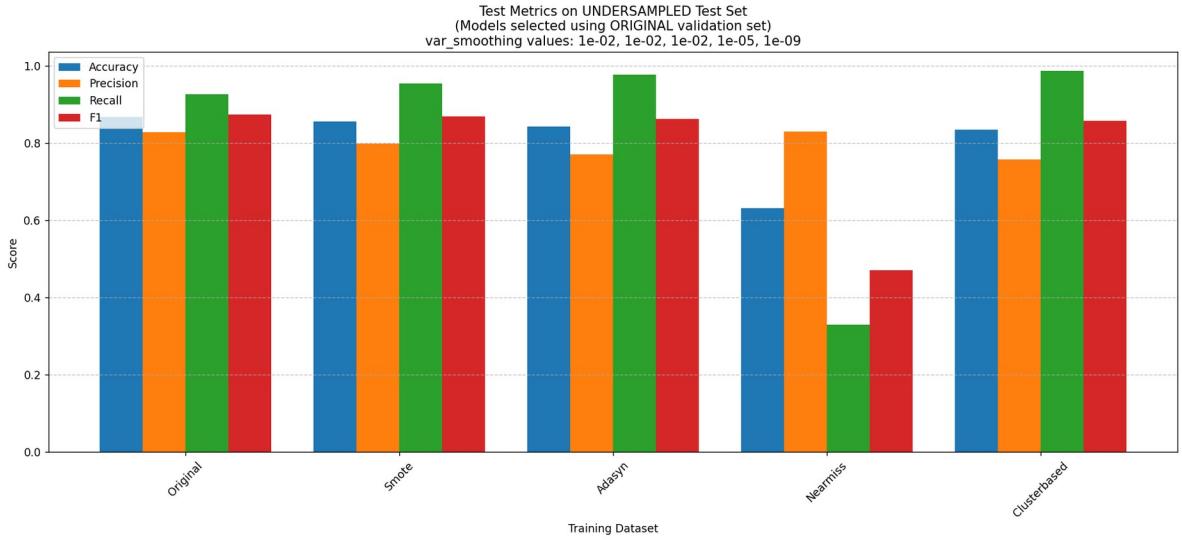

Validation Accuracy Comparison - Original Validation Set
Original
Smote
Adasyn
Nearmiss
Clusterbased
Validation Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1e-09
1e-08
1e-07
1e-06
1e-05
1e-04
1e-03
1e-02

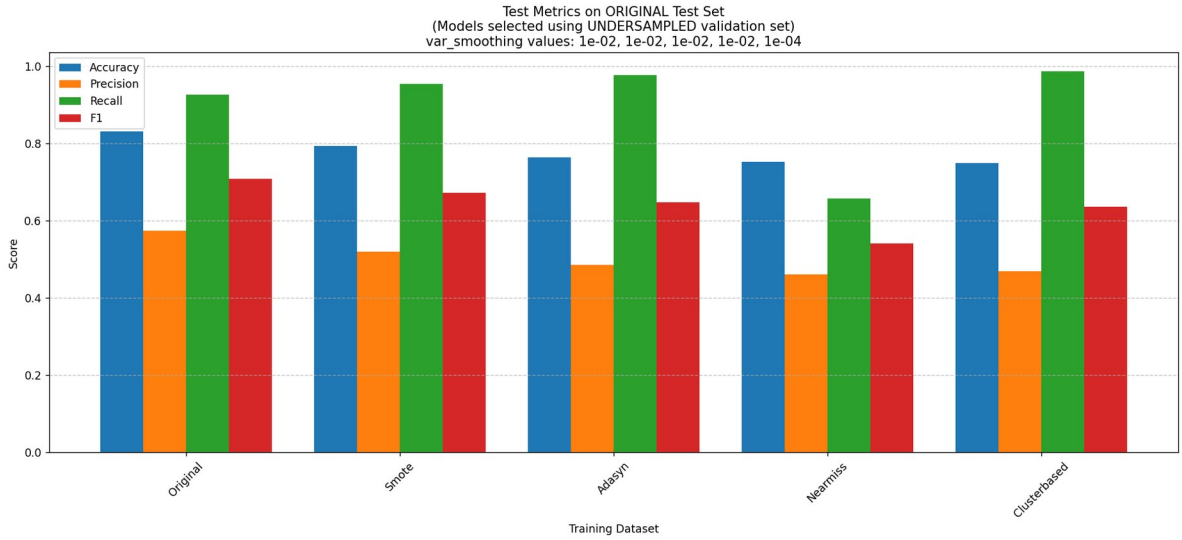

Validation Accuracy Heatmap - Original Validation Set
Training Set
original
smote
adasyn
nearmiss
clusterbased
0.740 0.740 0.741 0.743 0.747 0.760 0.784 0.816
0.741 0.742 0.743 0.744 0.746 0.749 0.759 0.778
0.726 0.728 0.731 0.735 0.740 0.742 0.742 0.745
0.784 0.790 0.794 0.800 0.801 0.794 0.775 0.759
0.735 0.735 0.735 0.735 0.735 0.735 0.735 0.734
1e-09
1e-08
1e-07
1e-06
1e-05
1e-04
1e-03
1e-02
var_smoothing Values
0.81
0.80
0.79
0.78
0.77
0.76
0.75
0.74
0.73

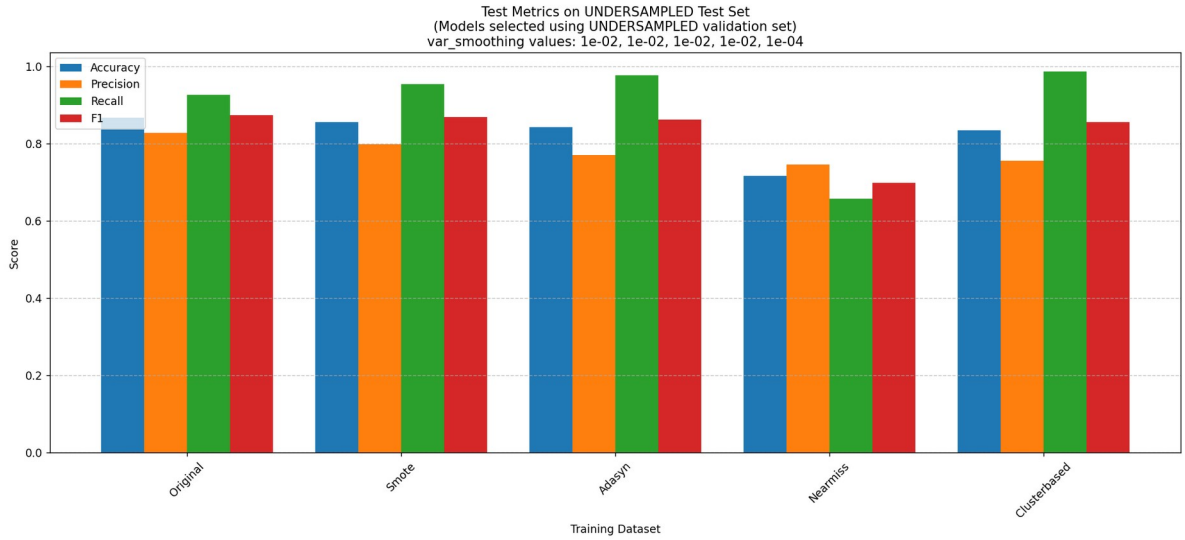

Validation Accuracy Comparison - Undersampled Validation Set
Original
Smote
Adasyn
Nearmiss
Clusterbased
Validation Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1e-09
1e-08
1e-07
1e-06
1e-05
1e-04
1e-03
1e-02


Validation Accuracy Heatmap - Undersampled Validation Set
Training Set
original
smote
adasyn
nearmiss
clusterbased
0.822 0.822 0.823 0.824 0.826 0.833 0.841 0.853
0.822 0.822 0.823 0.823 0.823 0.825 0.834 0.841
0.818 0.819 0.822 0.823 0.824 0.826 0.825 0.829
0.532 0.549 0.569 0.600 0.634 0.665 0.691 0.718
0.823 0.823 0.823 0.823 0.823 0.824 0.824 0.824
1e-09 1e-08 1e-07 1e-06 1e-05 1e-04 1e-03 1e-02
var_smoothing Values
0.85
0.80
0.75
0.70
0.65
0.60
0.55

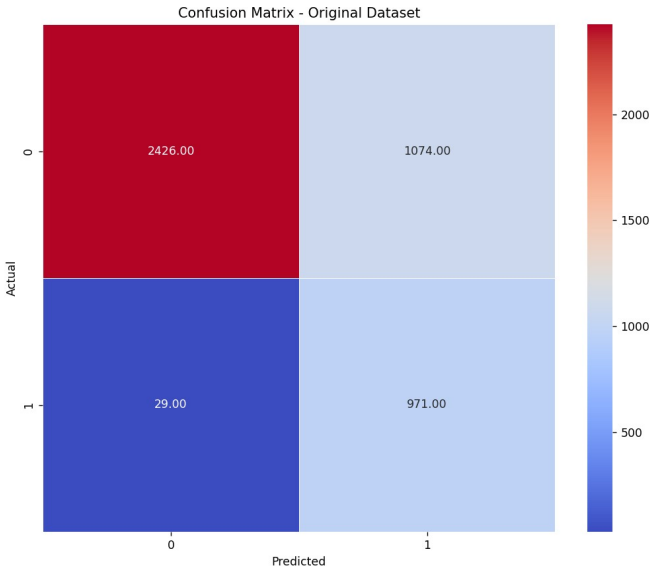

Test Metrics on ORIGINAL Test Set
(Models selected using ORIGINAL validation set)
var_smoothing values: 1e-02, 1e-02, 1e-02, 1e-05, 1e-09
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1
Original
Smote
Adasyn
Nearmiss
Clusterbased
Training Dataset


Test Metrics on UNDERSAMPLED Test Set
(Models selected using ORIGINAL validation set)
var_smoothing values: 1e-02, 1e-02, 1e-02, 1e-05, 1e-09
Accuracy
Precision
Recall
F1
Score
1.0
0.8
0.6
0.4
0.2
0.0
Original
Smote
Adasyn
Nearmiss
Clusterbased
Training Dataset


Test Metrics on ORIGINAL Test Set
(Models selected using UNDERSAMPLED validation set)
var_smoothing values: 1e-02, 1e-02, 1e-02, 1e-02, 1e-04
Accuracy
Precision
Recall
F1
Score
1.0
0.8
0.6
0.4
0.2
0.0
Original
Smote
Adasyn
Nearmiss
Clusterbased
Training Dataset



After here results were created without smoothing values and with original validation and test sets:

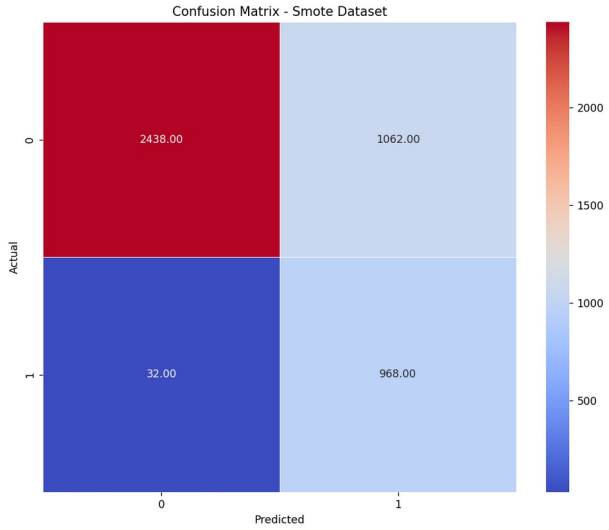


Validation Classification Report:

	precision	recall	f1-score	support
0	0.99	0.67	0.80	7000
1	0.46	0.97	0.62	2000
accuracy			0.74	9000
macro avg	0.72	0.82	0.71	9000
weighted avg	0.87	0.74	0.76	9000

Test Classification Report:

	precision	recall	f1-score	support
0	0.99	0.69	0.81	3500
1	0.47	0.97	0.64	1000
accuracy			0.75	4500
macro avg	0.73	0.83	0.73	4500
weighted avg	0.87	0.75	0.78	4500

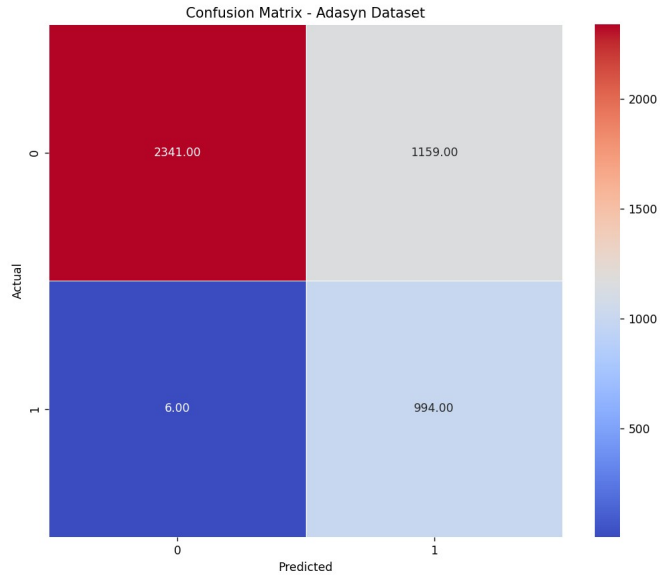


Validation Classification Report:

	precision	recall	f1-score	support
0	0.99	0.67	0.80	7000
1	0.46	0.97	0.62	2000
accuracy			0.74	9000
macro avg	0.72	0.82	0.71	9000
weighted avg	0.87	0.74	0.76	9000

Test Classification Report:

	precision	recall	f1-score	support
0	0.99	0.70	0.82	3500
1	0.48	0.97	0.64	1000
accuracy			0.76	4500
macro avg	0.73	0.83	0.73	4500
weighted avg	0.87	0.76	0.78	4500

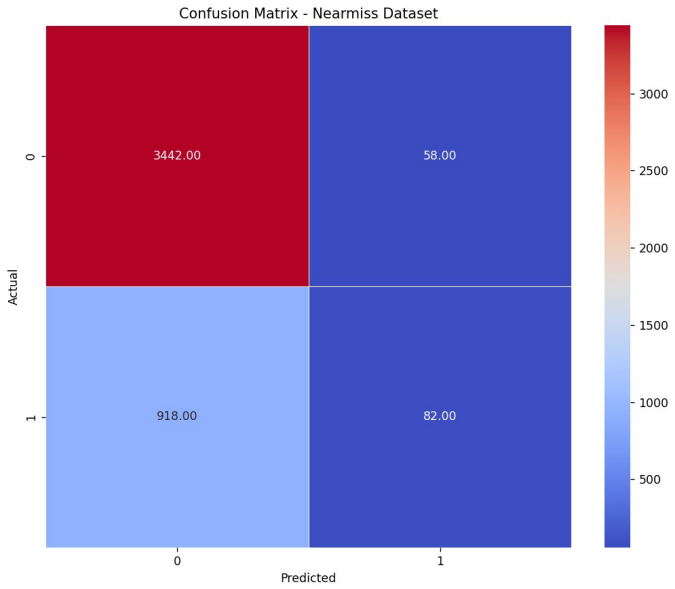


Validation Classification Report:

	precision	recall	f1-score	support
0	1.00	0.65	0.79	7000
1	0.45	0.99	0.62	2000
accuracy			0.73	9000
macro avg	0.72	0.82	0.70	9000
weighted avg	0.88	0.73	0.75	9000

Test Classification Report:

	precision	recall	f1-score	support
0	1.00	0.67	0.80	3500
1	0.46	0.99	0.63	1000
accuracy			0.74	4500
macro avg	0.73	0.83	0.72	4500
weighted avg	0.88	0.74	0.76	4500

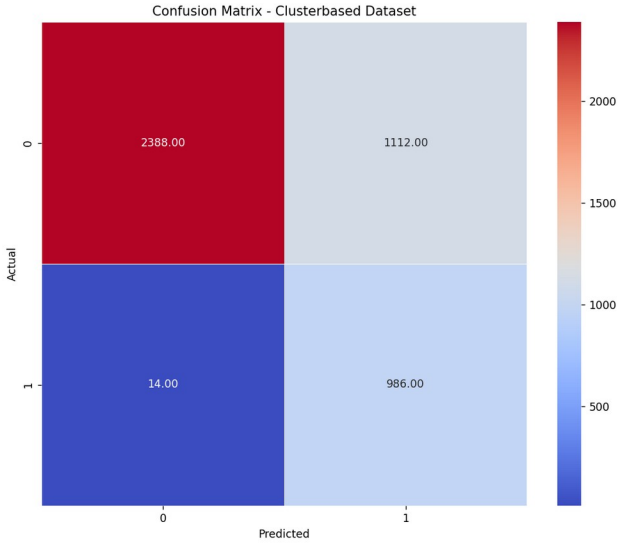


Validation Classification Report:

	precision	recall	f1-score	support
0	0.79	0.98	0.88	7000
1	0.60	0.08	0.14	2000
accuracy			0.78	9000
macro avg	0.69	0.53	0.51	9000
weighted avg	0.75	0.78	0.71	9000

Test Classification Report:

	precision	recall	f1-score	support
0	0.79	0.98	0.88	3500
1	0.59	0.08	0.14	1000
accuracy			0.78	4500
macro avg	0.69	0.53	0.51	4500
weighted avg	0.74	0.78	0.71	4500



Validation Classification Report:

	precision	recall	f1-score	support
0	1.00	0.66	0.80	7000
1	0.46	0.99	0.62	2000
accuracy			0.73	9000
macro avg	0.73	0.83	0.71	9000
weighted avg	0.88	0.73	0.76	9000

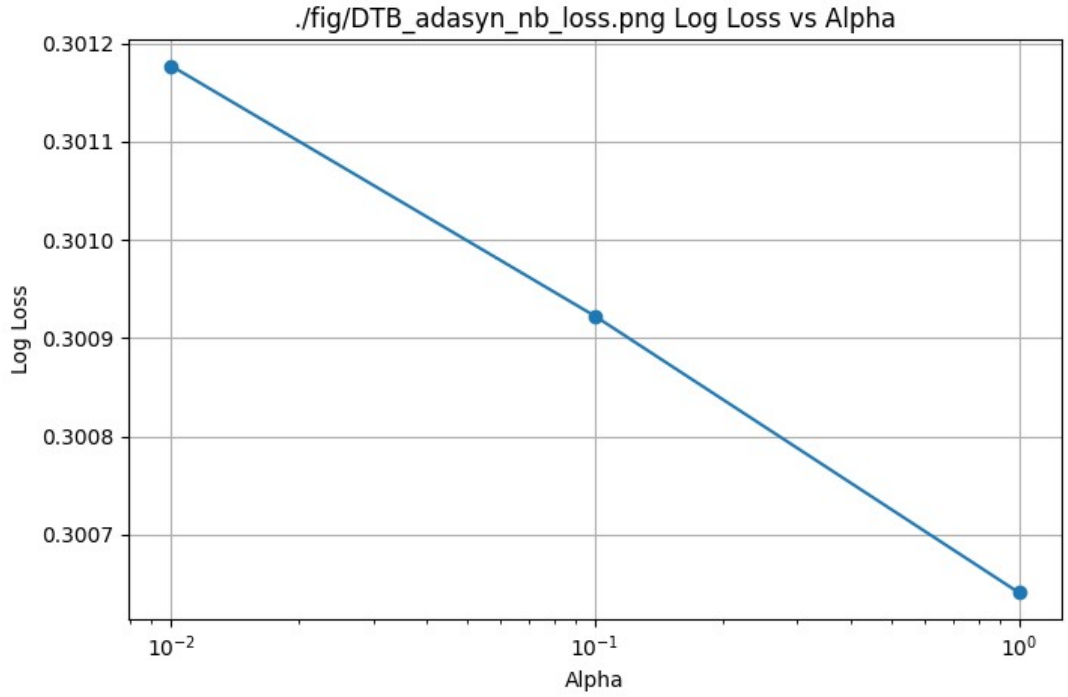
Test Classification Report:

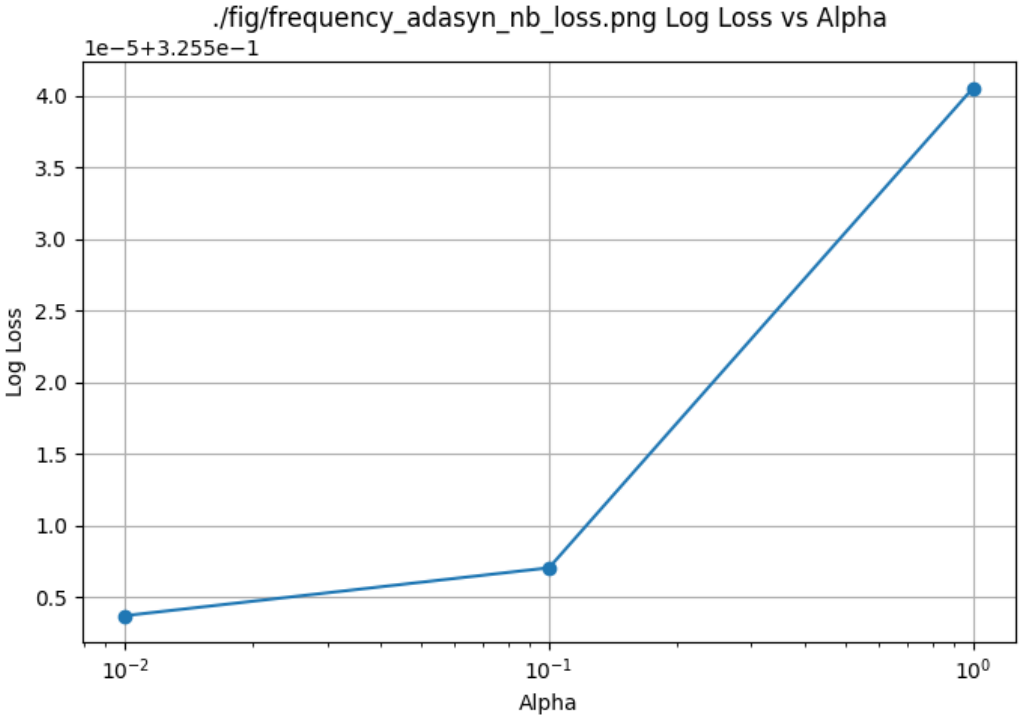
	precision	recall	f1-score	support
0	0.99	0.68	0.81	3500
1	0.47	0.99	0.64	1000
accuracy			0.75	4500
macro avg	0.73	0.83	0.72	4500
weighted avg	0.88	0.75	0.77	4500

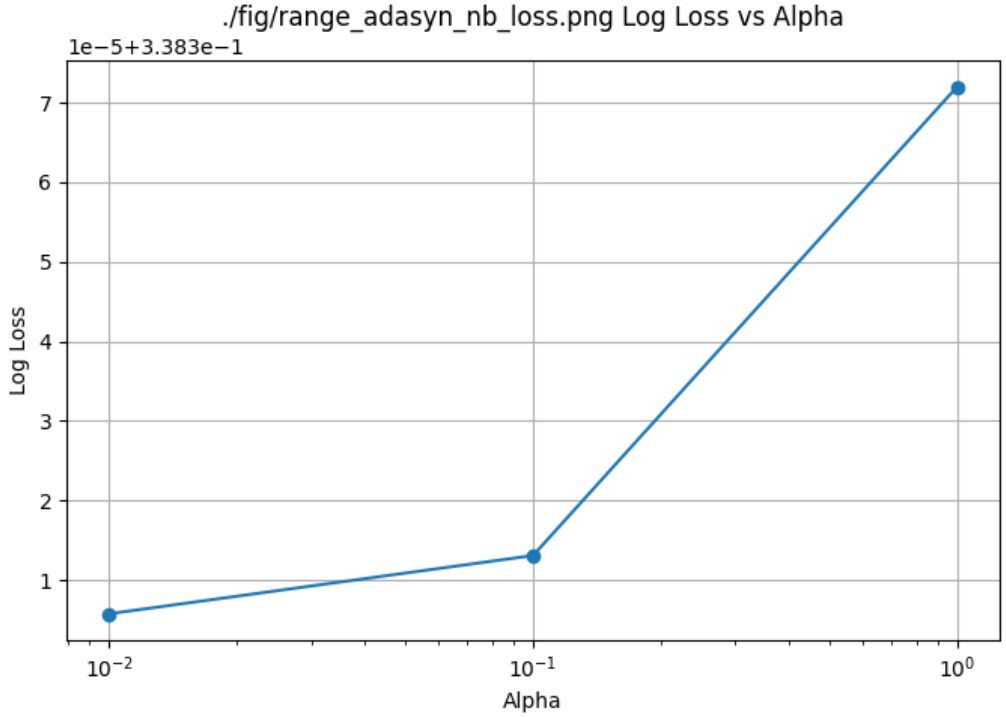
Naive Bayes

We implemented our own Naive Bayes classification algorithm which allows us to classify data based on calculated probabilities. We started by determining the prior probabilities for each class and the likelihoods of feature values given these classes. Using this information we made predictions about the class of new data instances. Our approach also provides flexibility by including features to save and load trained models, which makes it easier to reuse the models.

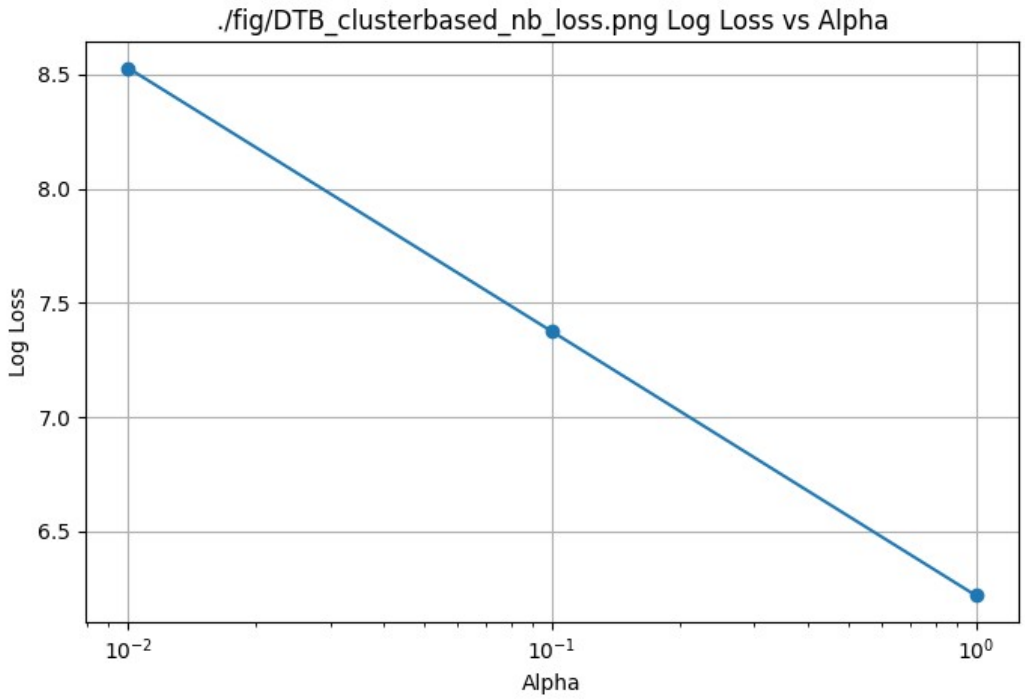
Adasyn:

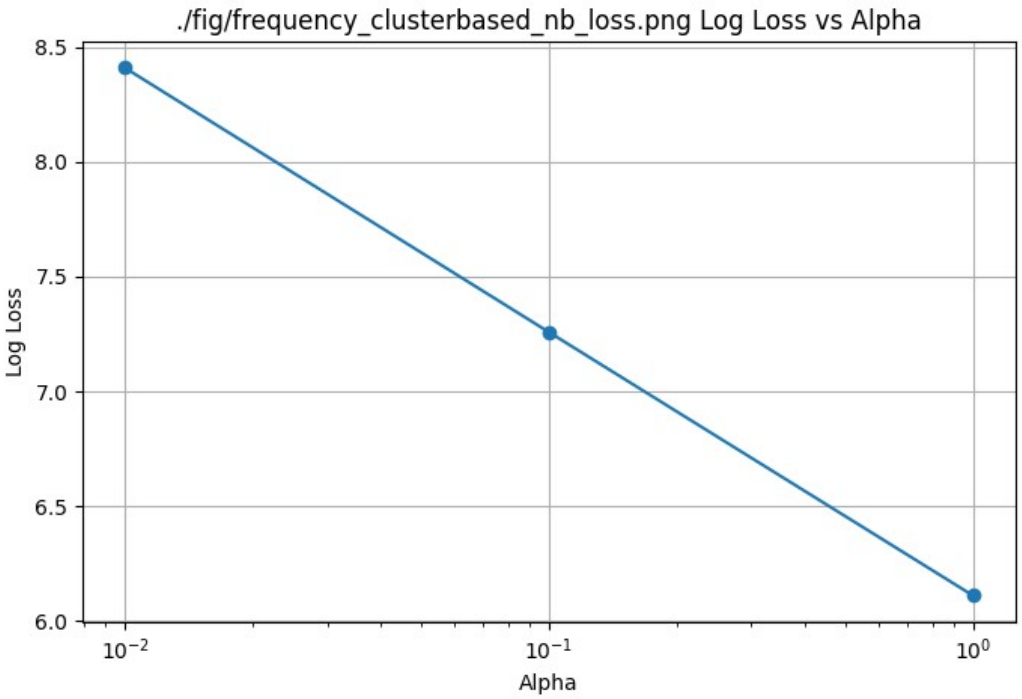


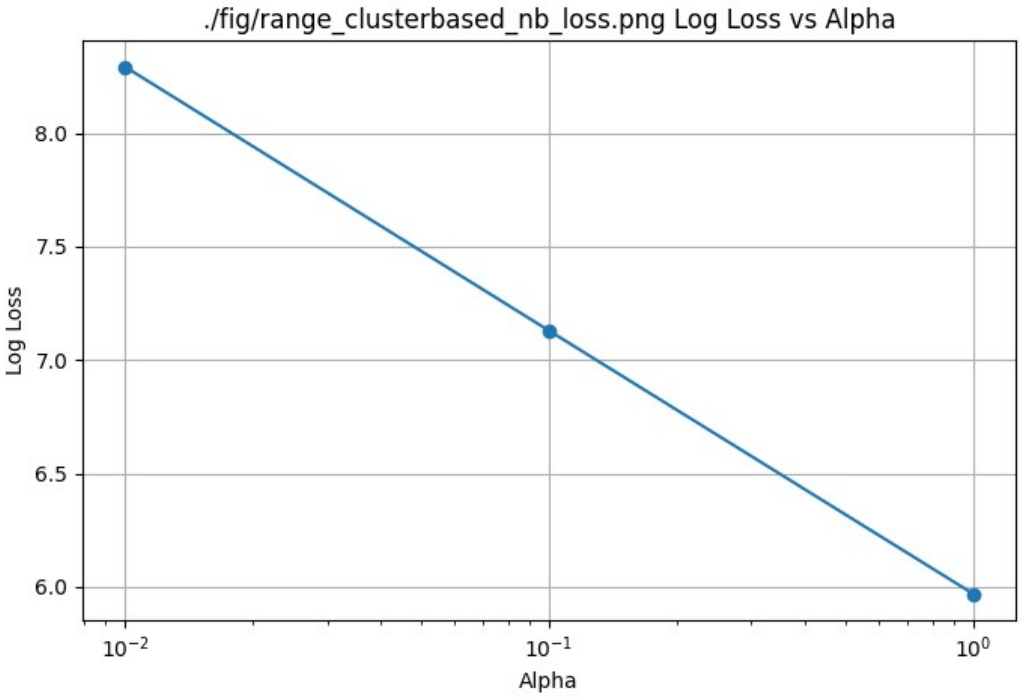

./fig/frequency_adasyn_nb_loss.png Log Loss vs Alpha
1e−5+3.255e−1
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Log Loss
10^{-2}
10^{-1}
10^{0}
Alpha


./fig/range_adasyn_nb_loss.png Log Loss vs Alpha
1e−5+3.383e−1
7
6
5
4
3
2
1
Log Loss
10^{-2}
10^{-1}
10^{0}
Alpha

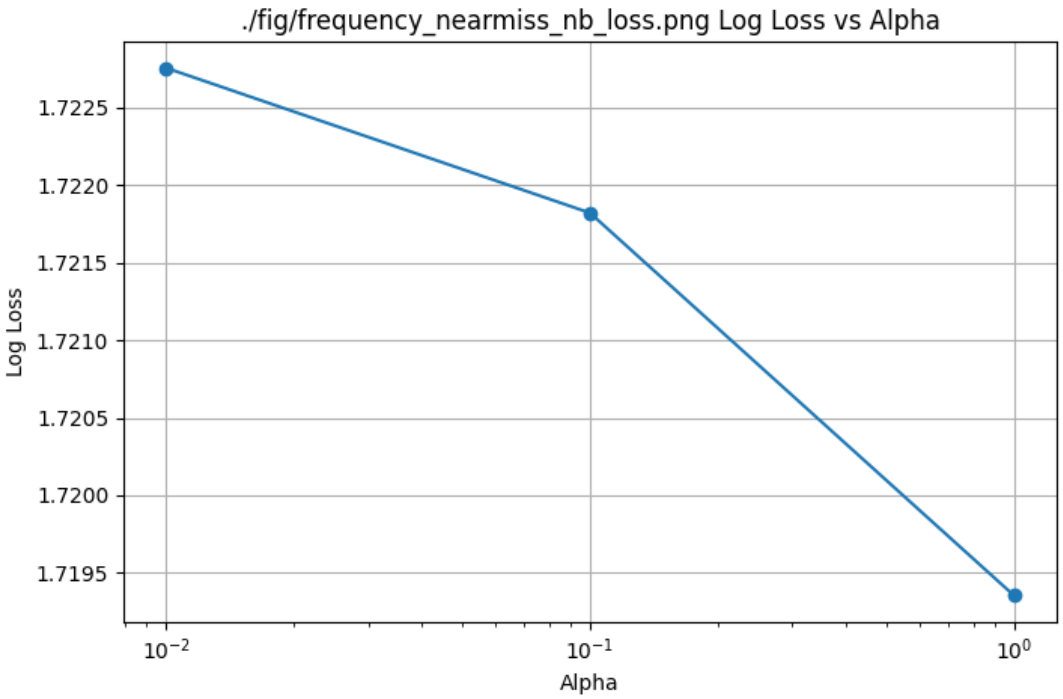
Clusterbased:

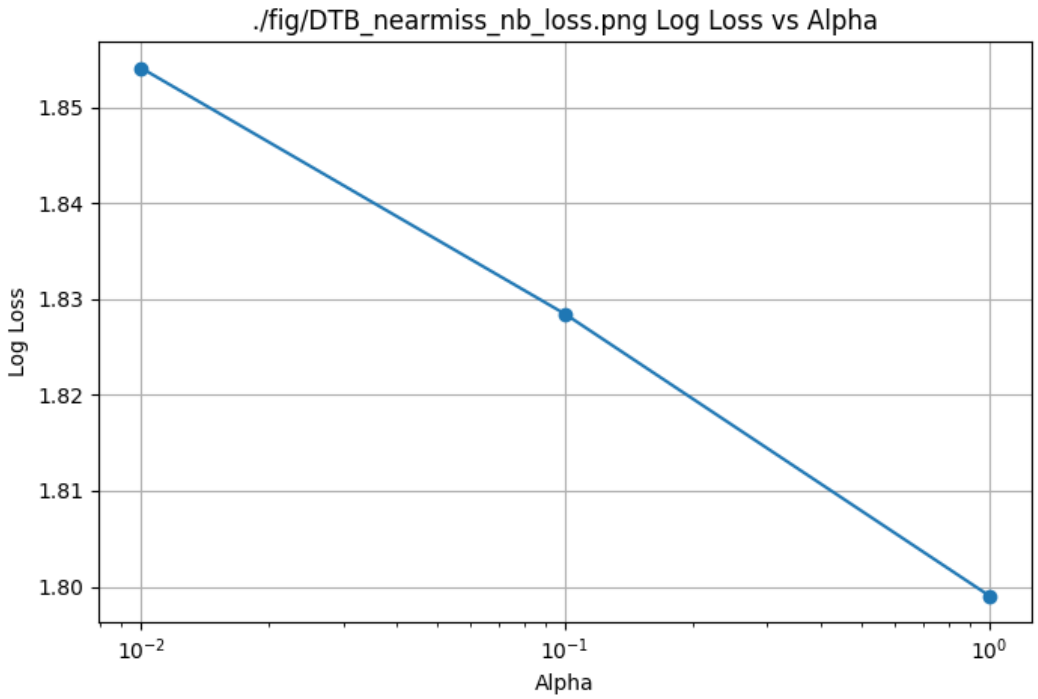


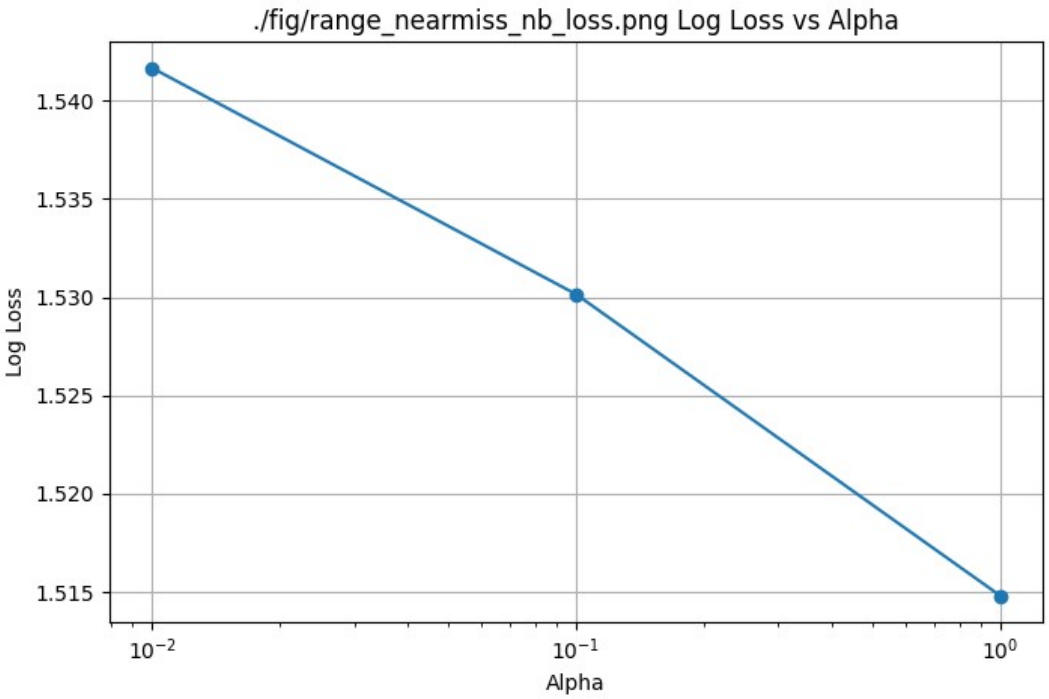

./fig/frequency_clusterbased_nb_loss.png Log Loss vs Alpha
Log Loss
Alpha
8.5
8.0
7.5
7.0
6.5
6.0
10^{-2}
10^{-1}
10^{0}


./fig/range_clusterbased_nb_loss.png Log Loss vs Alpha
Log Loss
Alpha
8.0
7.5
7.0
6.5
6.0
10^{-2}
10^{-1}
10^{0}

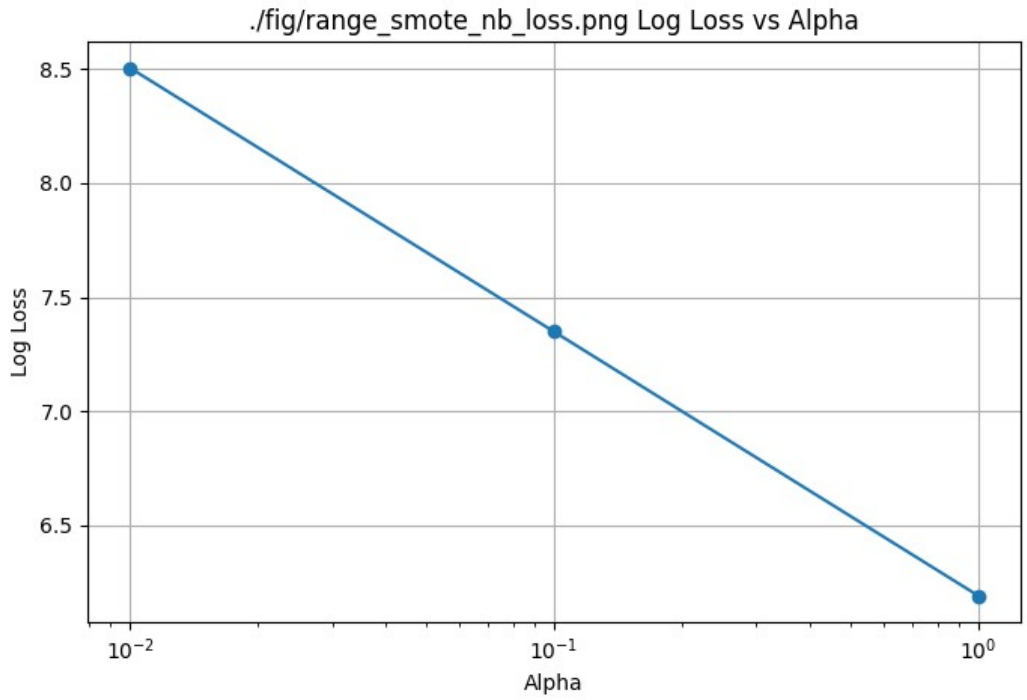
Nearmiss:

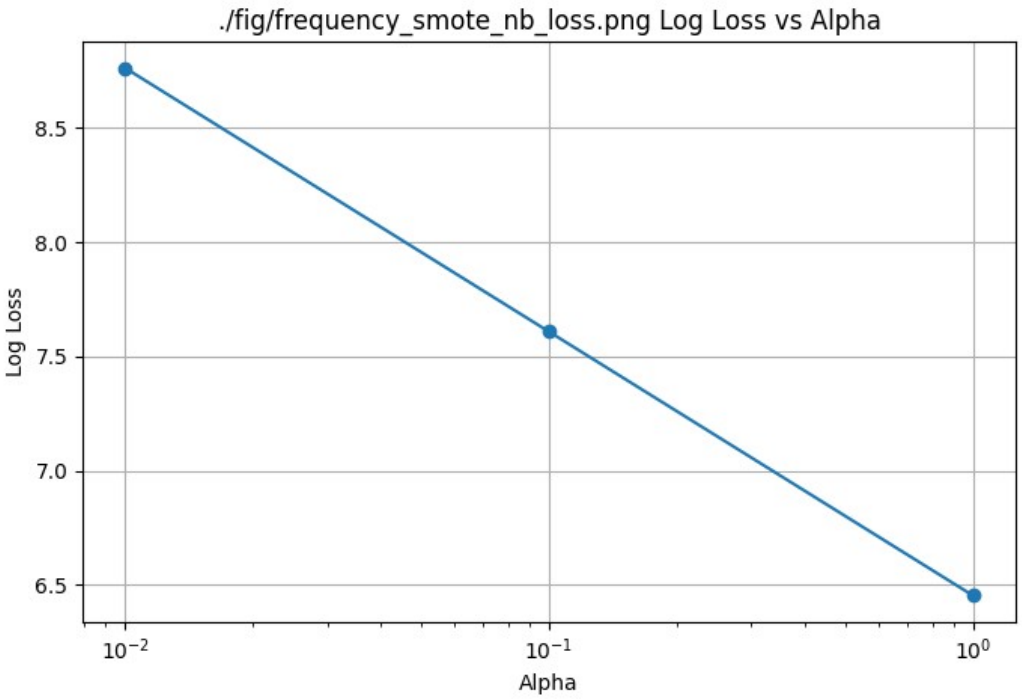

./fig/frequency_nearmiss_nb_loss.png Log Loss vs Alpha
Log Loss
1.7225
1.7220
1.7215
1.7210
1.7205
1.7200
1.7195
10^{-2}
10^{-1}
10^{0}
Alpha

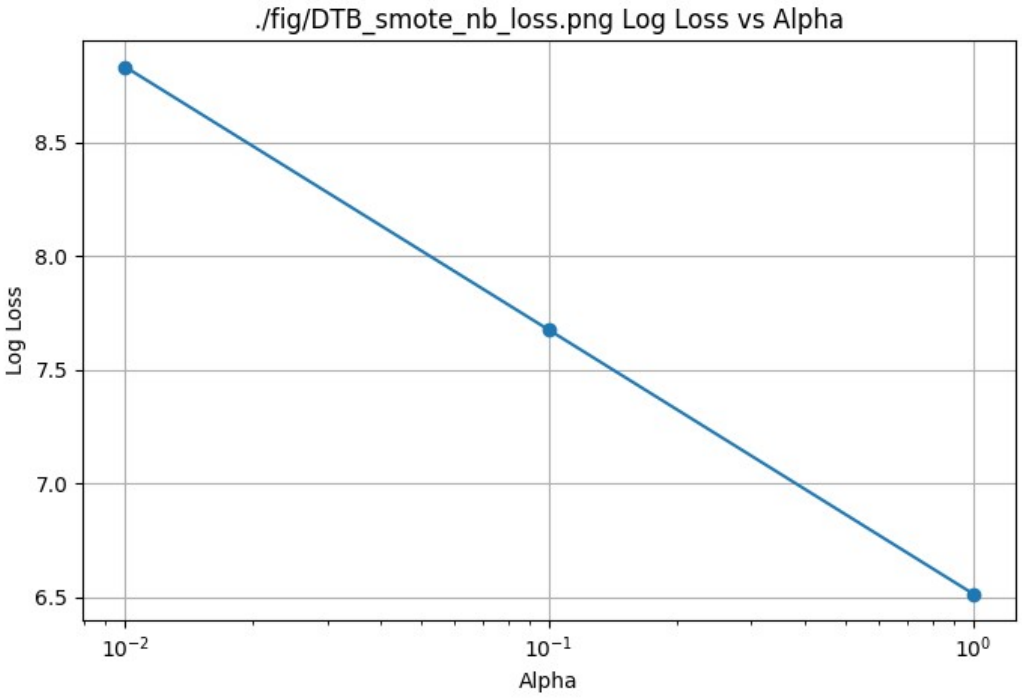

./fig/DTB_nearmiss_nb_loss.png Log Loss vs Alpha
Log Loss
1.85
1.84
1.83
1.82
1.81
1.80
10^{-2}
10^{-1}
10^{0}
Alpha


./fig/range_nearmiss_nb_loss.png Log Loss vs Alpha
Log Loss
1.540
1.535
1.530
1.525
1.520
1.515
10^{-2}
10^{-1}
10^{0}
Alpha

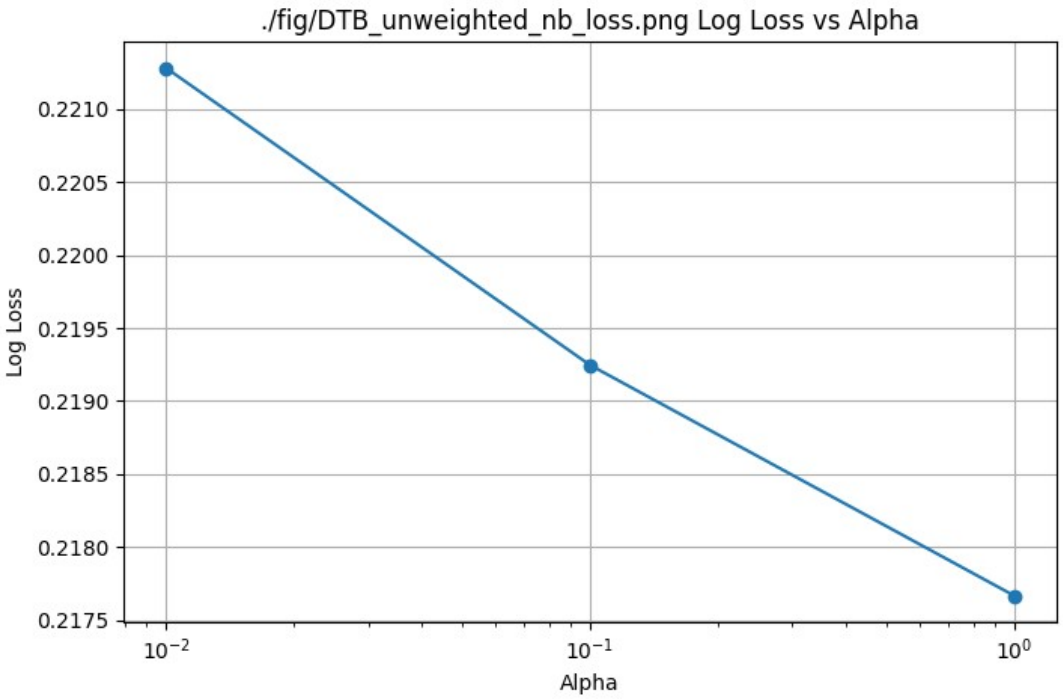
SMOTE:

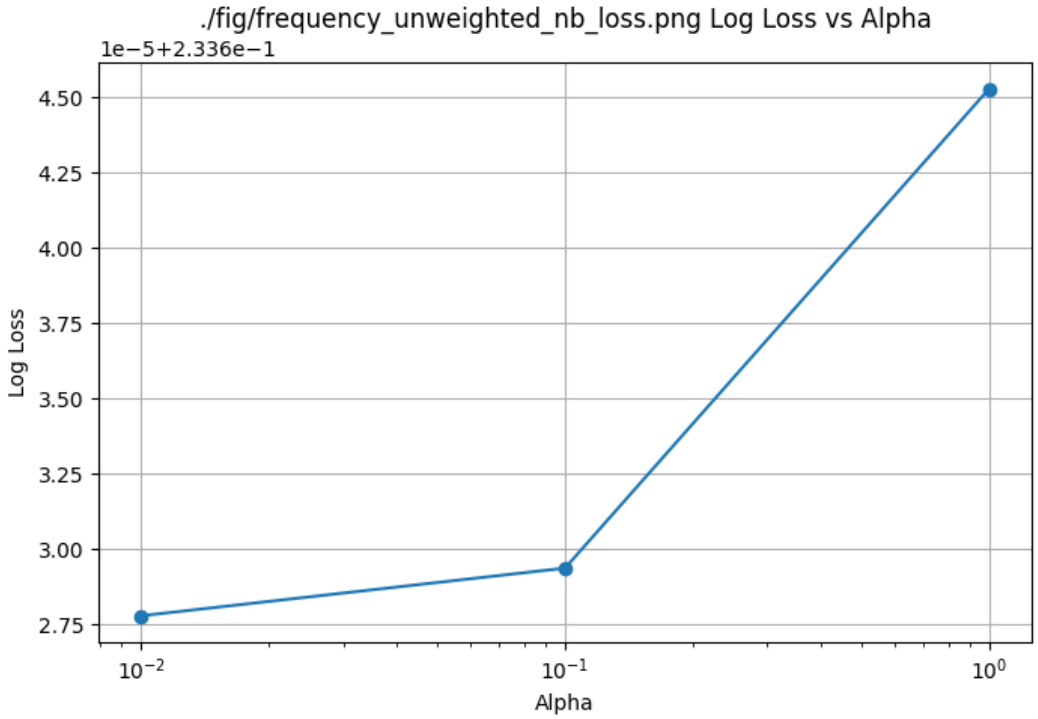

./fig/range_smote_nb_loss.png Log Loss vs Alpha
8.5
8.0
7.5
7.0
6.5
Log Loss
10^{-2}
10^{-1}
10^{0}
Alpha

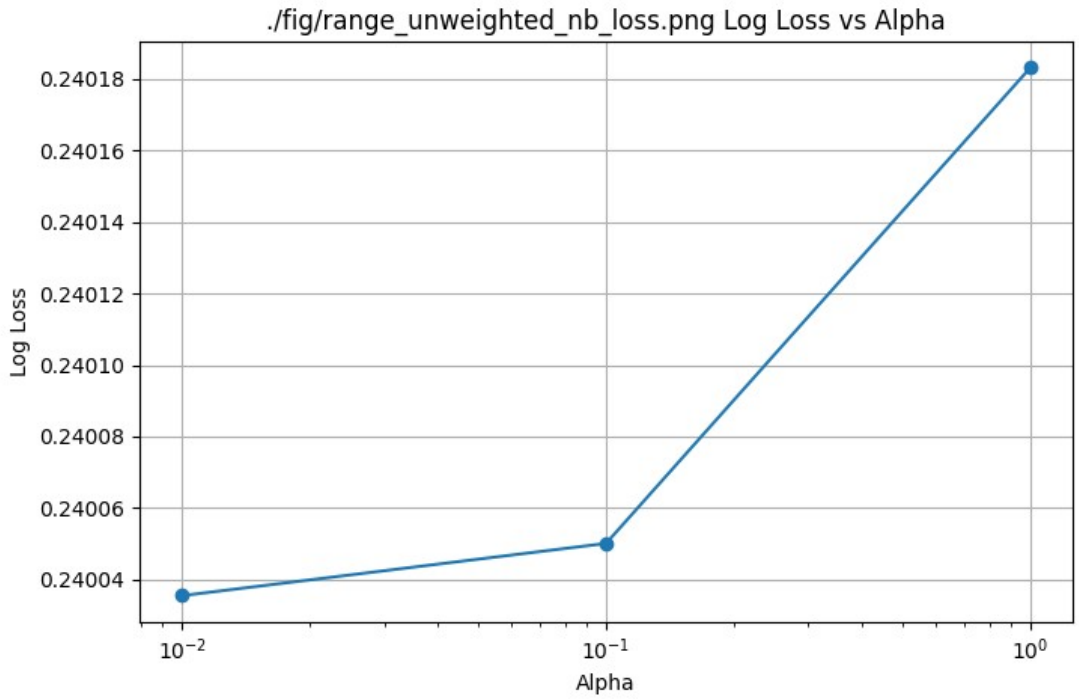

./fig/frequency_smote_nb_loss.png Log Loss vs Alpha
Log Loss
8.5
8.0
7.5
7.0
6.5
10^{-2}
10^{-1}
10^{0}
Alpha


./fig/DTB_smote_nb_loss.png Log Loss vs Alpha
Log Loss
8.5
8.0
7.5
7.0
6.5
10^{-2}
10^{-1}
10^{0}
Alpha

Unweighted:


./fig/DTB_unweighted_nb_loss.png Log Loss vs Alpha
0.2210
0.2205
0.2200
0.2195
0.2190
0.2185
0.2180
0.2175
Log Loss
10^{-2}
10^{-1}
10^{0}
Alpha

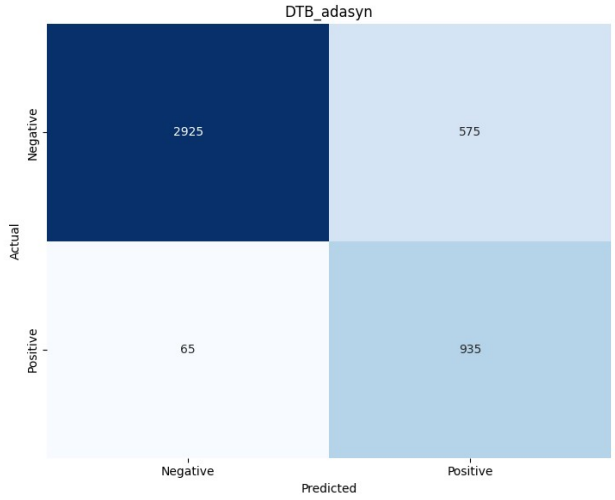

./fig/frequency_unweighted_nb_loss.png Log Loss vs Alpha
1e−5+2.336e−1
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
Log Loss
10^{-2}
10^{-1}
10^{0}
Alpha



Confusion Matrices For Naive Bayes

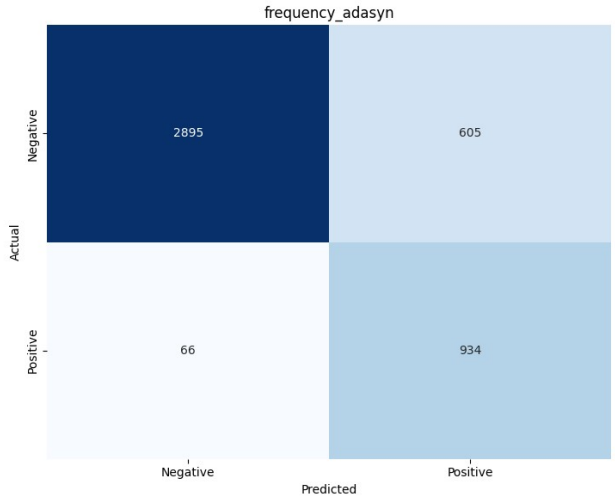
Original Test and Validation Sets

Adasyn:



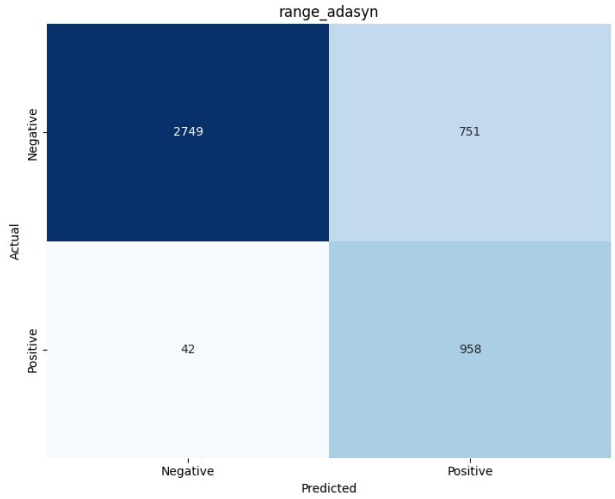
Metrics

Accuracy: 0.86
Precision: 0.62
Recall: 0.94
Specificity: 0.84
F1 Score: 0.75



Metrics

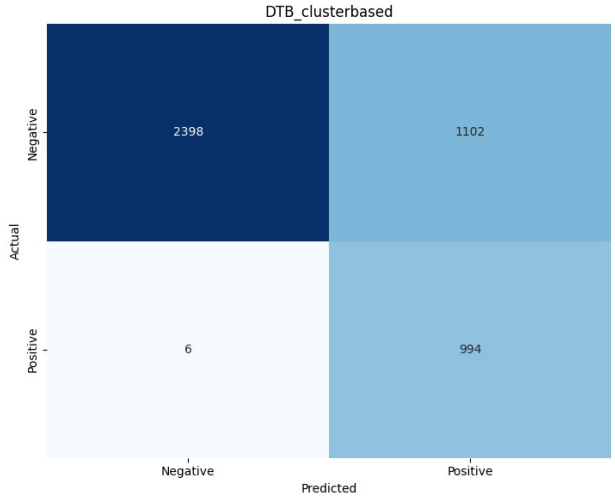
Accuracy: 0.85
Precision: 0.61
Recall: 0.93
Specificity: 0.83
F1 Score: 0.74



Metrics

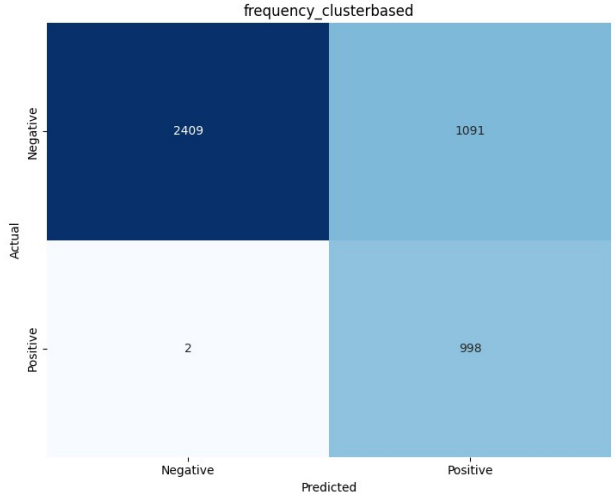
Accuracy: 0.82
Precision: 0.56
Recall: 0.96
Specificity: 0.79
F1 Score: 0.71

Clusterbased:



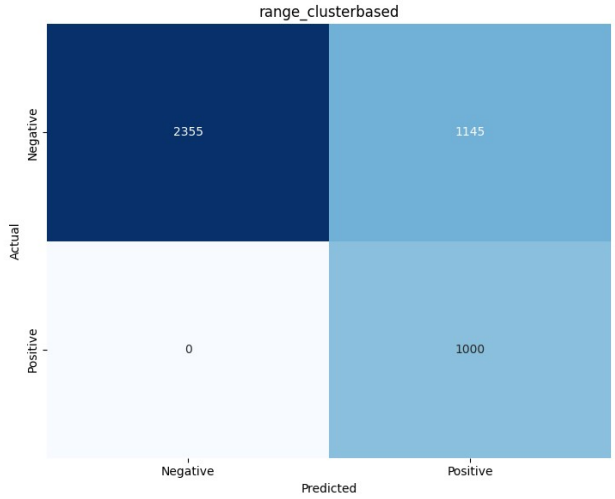
Metrics

Accuracy: 0.75
Precision: 0.47
Recall: 0.99
Specificity: 0.69
F1 Score: 0.64



Metrics

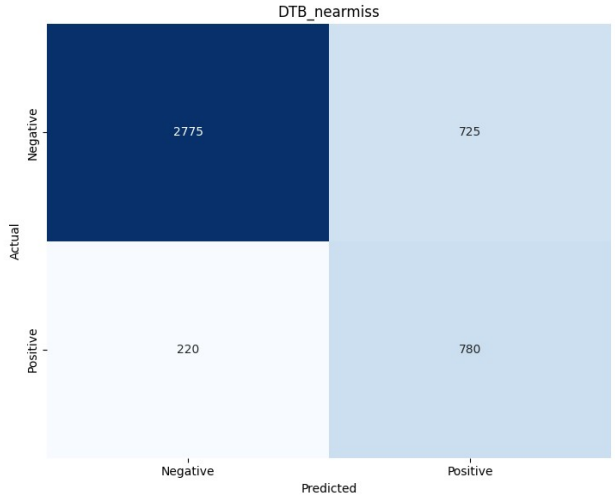
Accuracy: 0.76
Precision: 0.48
Recall: 1.00
Specificity: 0.69
F1 Score: 0.65



Metrics

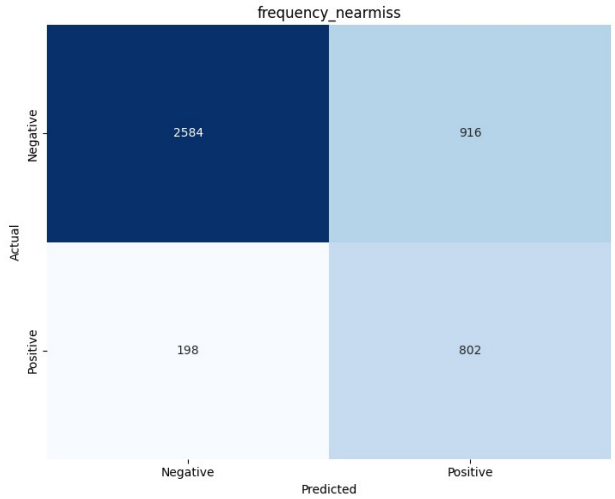
Accuracy: 0.75
Precision: 0.47
Recall: 1.00
Specificity: 0.67
F1 Score: 0.64

Nearmiss:



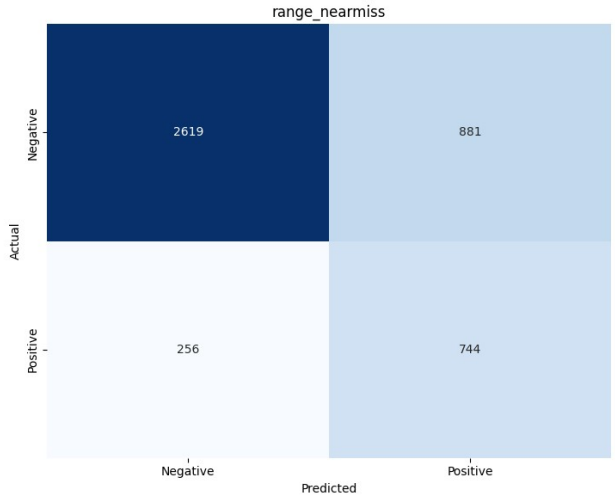
Metrics

Accuracy: 0.79
Precision: 0.52
Recall: 0.78
Specificity: 0.79
F1 Score: 0.62



Metrics

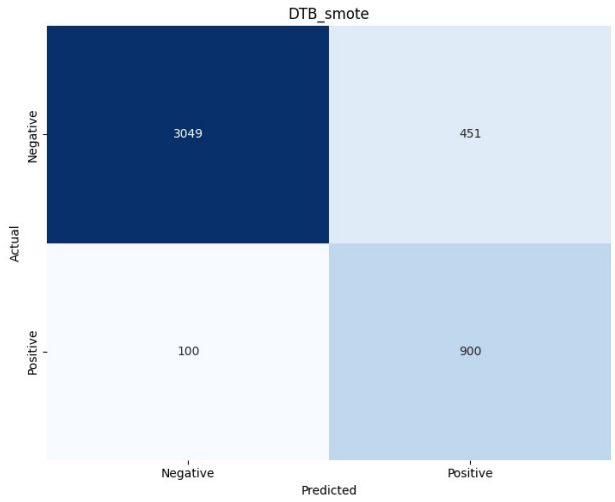
Accuracy: 0.75
Precision: 0.47
Recall: 0.80
Specificity: 0.74
F1 Score: 0.59



Metrics

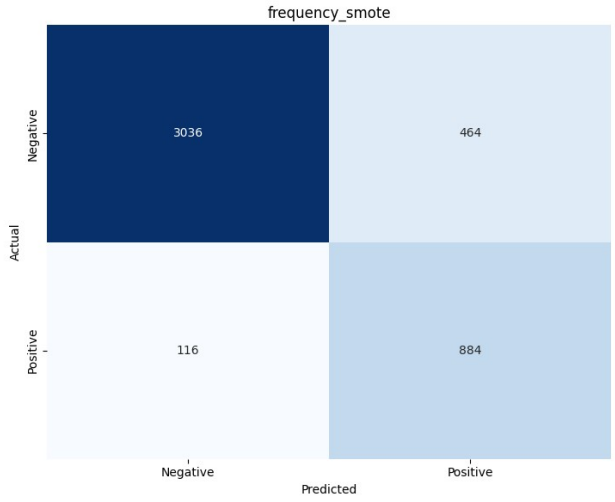
Accuracy: 0.75
Precision: 0.46
Recall: 0.74
Specificity: 0.75
F1 Score: 0.57

SMOTE:



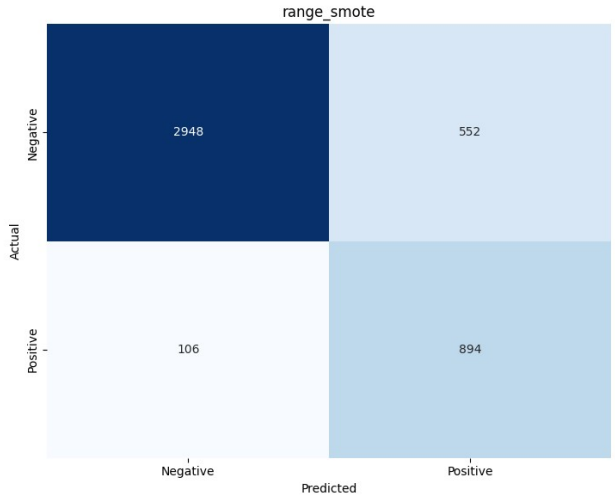
Metrics

Accuracy: 0.88
Precision: 0.67
Recall: 0.90
Specificity: 0.87
F1 Score: 0.77



Metrics

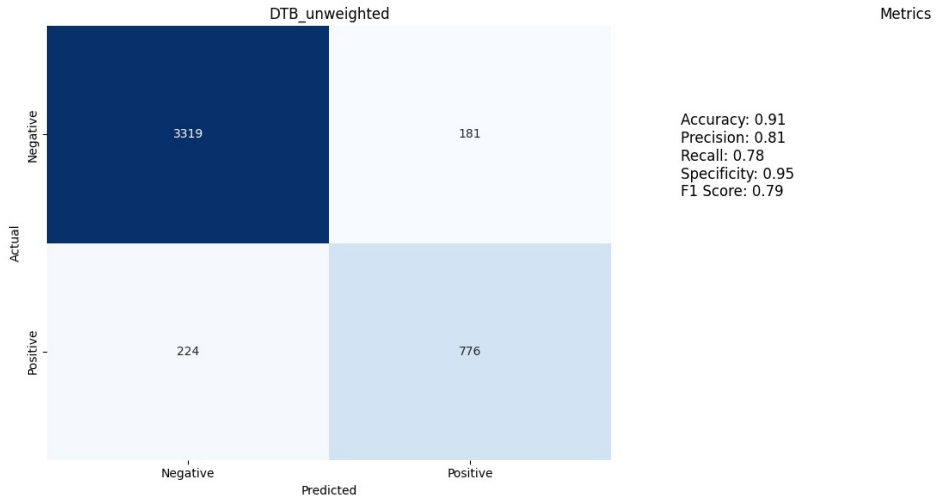
Accuracy: 0.87
Precision: 0.66
Recall: 0.88
Specificity: 0.87
F1 Score: 0.75

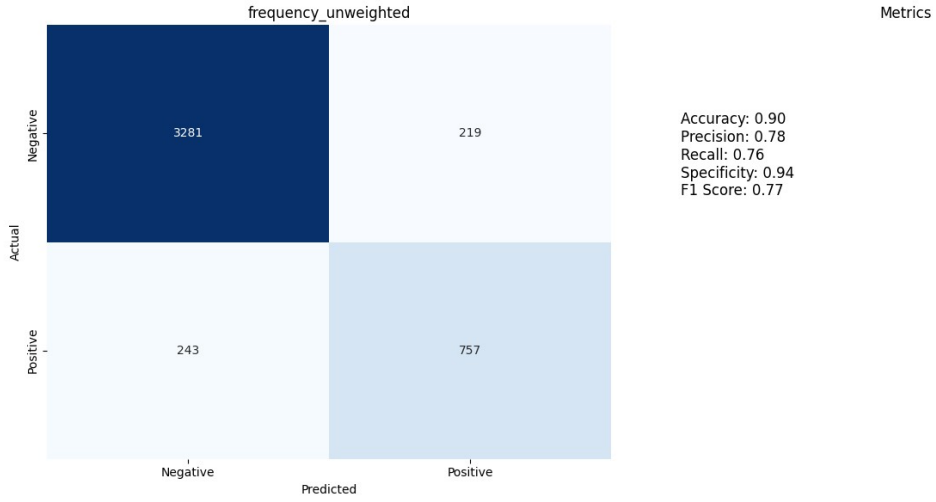


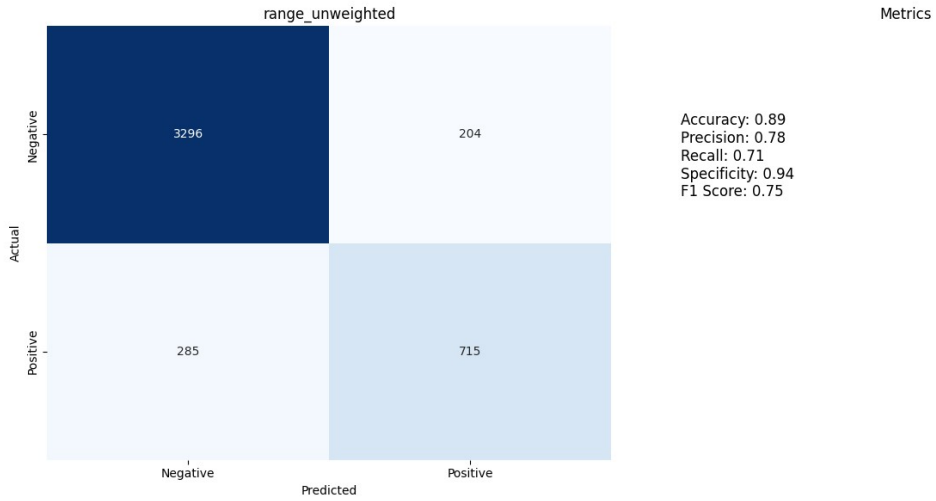
Metrics

Accuracy: 0.85
Precision: 0.62
Recall: 0.89
Specificity: 0.84
F1 Score: 0.73

Unweighted:

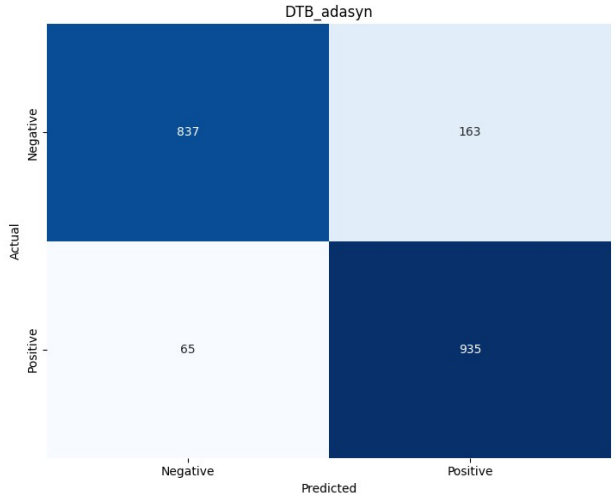






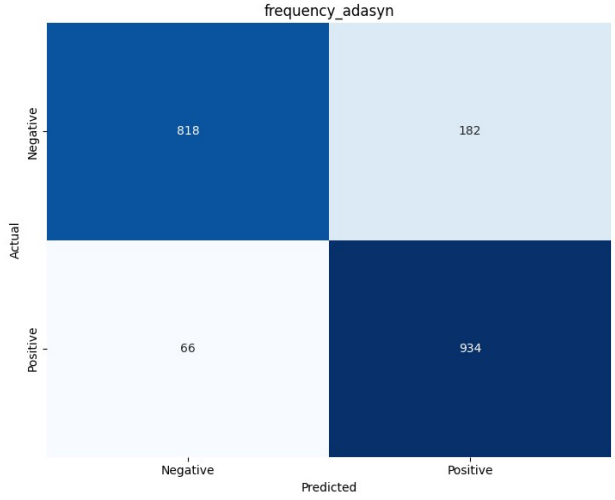
Undersampled Test And Validation Sets

Adasyn:



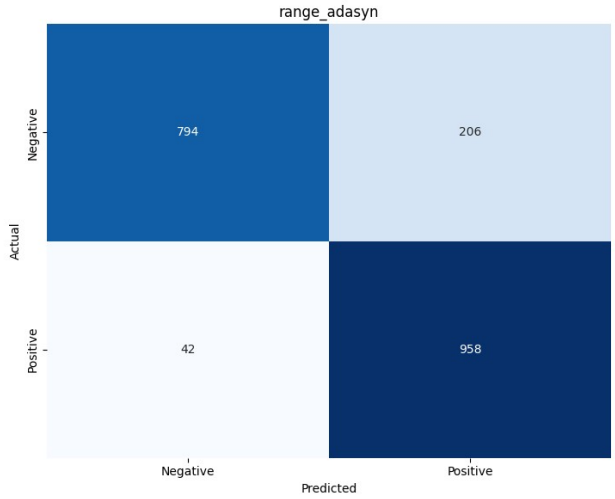
Metrics

Accuracy: 0.89
Precision: 0.85
Recall: 0.94
Specificity: 0.84
F1 Score: 0.89



Metrics

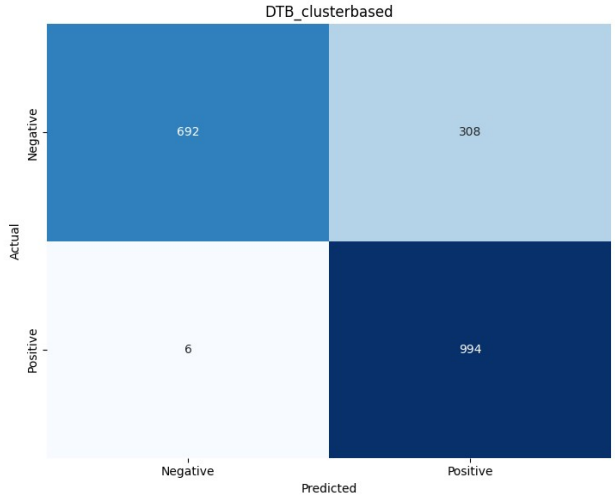
Accuracy: 0.88
Precision: 0.84
Recall: 0.93
Specificity: 0.82
F1 Score: 0.88



Metrics

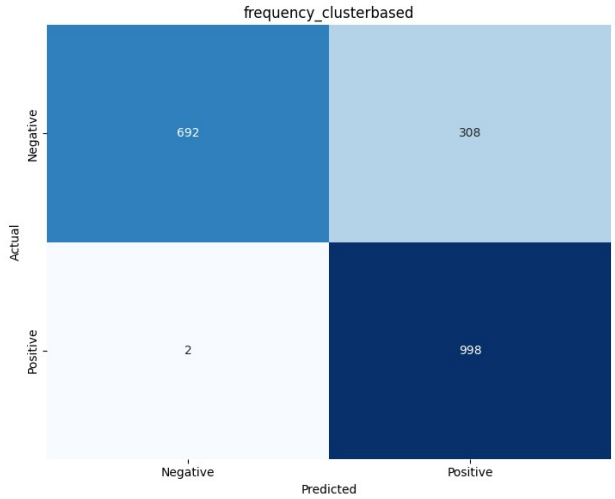
Accuracy: 0.88
Precision: 0.82
Recall: 0.96
Specificity: 0.79
F1 Score: 0.89

Clusterbased:



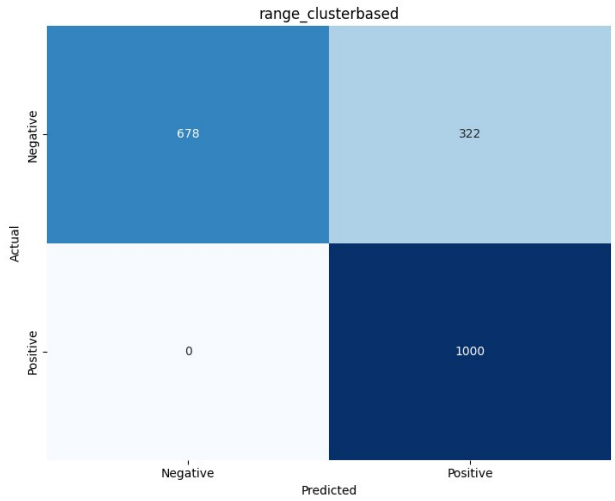
Metrics

Accuracy: 0.84
Precision: 0.76
Recall: 0.99
Specificity: 0.69
F1 Score: 0.86



Metrics

Accuracy: 0.84
Precision: 0.76
Recall: 1.00
Specificity: 0.69
F1 Score: 0.87

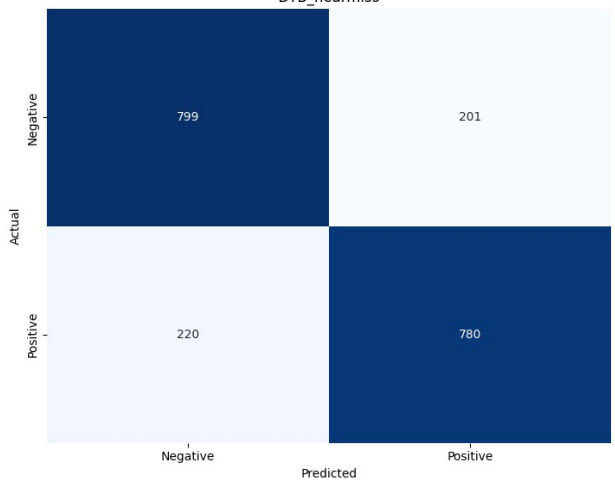


Metrics

Accuracy: 0.84
Precision: 0.76
Recall: 1.00
Specificity: 0.68
F1 Score: 0.86

Nearmiss:

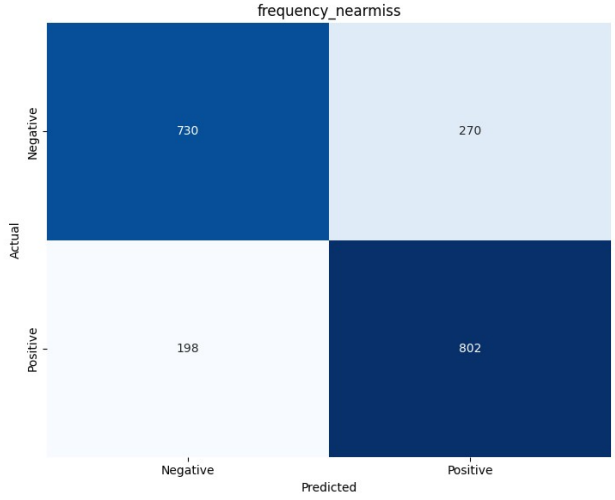
DTB_nearmiss



Metrics

Accuracy: 0.79
Precision: 0.80
Recall: 0.78
Specificity: 0.80
F1 Score: 0.79

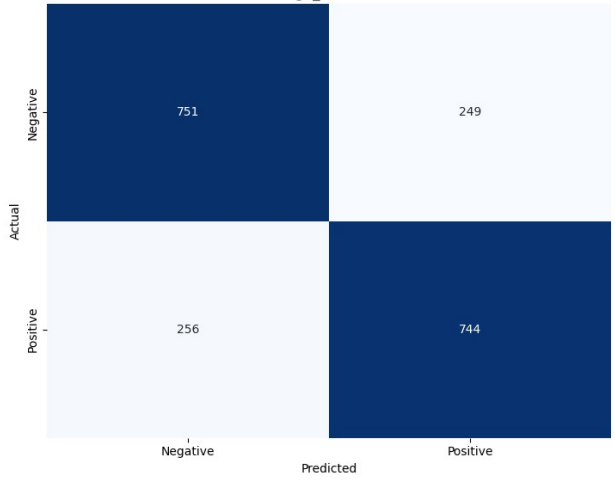
frequency_nearmiss



Metrics

Accuracy: 0.77
Precision: 0.75
Recall: 0.80
Specificity: 0.73
F1 Score: 0.77

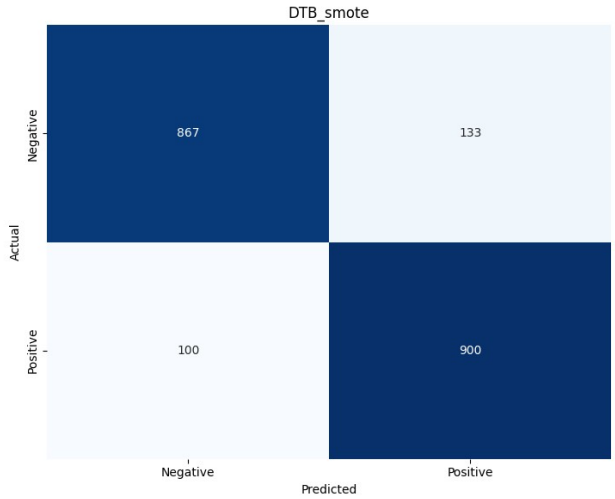
range_nearmiss



Metrics

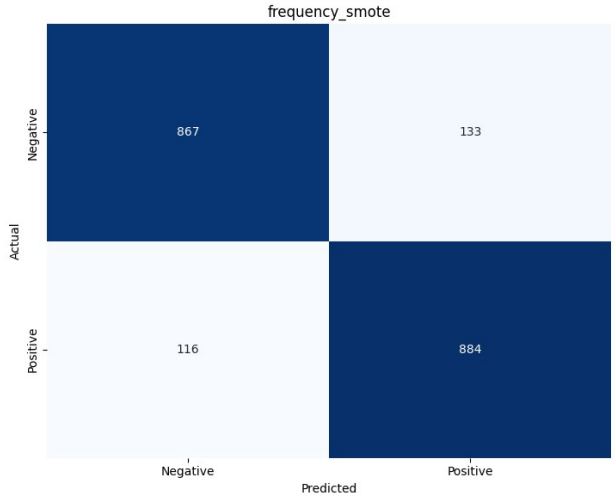
Accuracy: 0.75
Precision: 0.75
Recall: 0.74
Specificity: 0.75
F1 Score: 0.75

SMOTE:



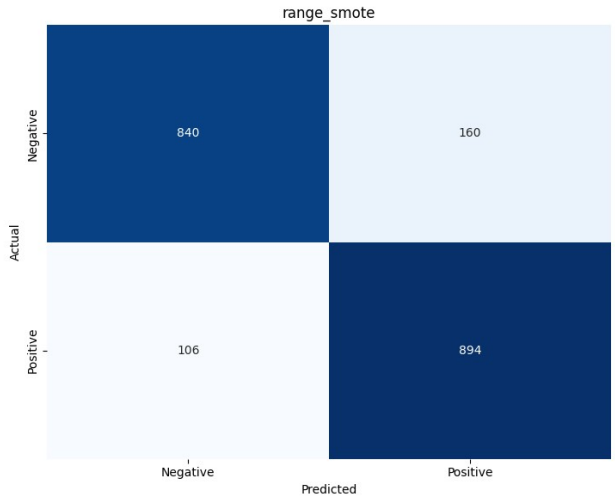
Metrics

Accuracy: 0.88
Precision: 0.87
Recall: 0.90
Specificity: 0.87
F1 Score: 0.89



Metrics

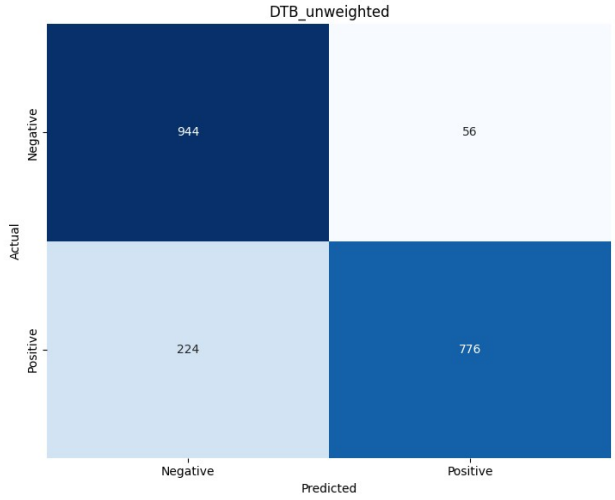
Accuracy: 0.88
Precision: 0.87
Recall: 0.88
Specificity: 0.87
F1 Score: 0.88



Metrics

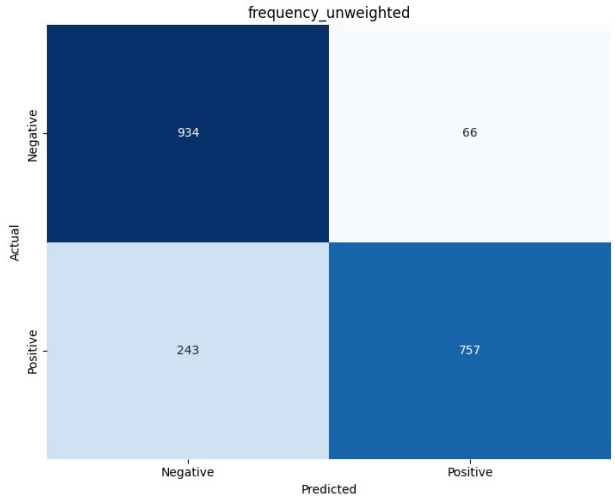
Accuracy: 0.87
Precision: 0.85
Recall: 0.89
Specificity: 0.84
F1 Score: 0.87

Unweighted:



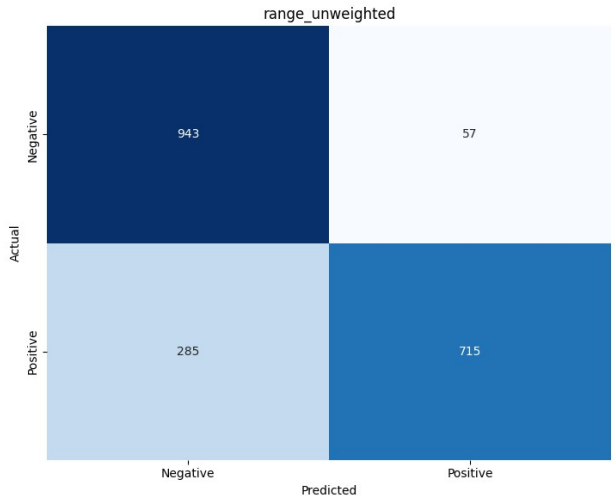
Metrics

Accuracy: 0.86
Precision: 0.93
Recall: 0.78
Specificity: 0.94
F1 Score: 0.85



Metrics

Accuracy: 0.85
Precision: 0.92
Recall: 0.76
Specificity: 0.93
F1 Score: 0.83



Metrics

Accuracy: 0.83
Precision: 0.93
Recall: 0.71
Specificity: 0.94
F1 Score: 0.81

Adaboost Naive Bayes

Boosted Naive Bayes is an application of Adaboost using Naive Bayes as the base learner. It enhances the performance of Naive Bayes by addressing its limitations, such as sensitivity to feature independence assumptions, through iterative weighting and model combination.

- Initialization: The algorithm begins by assigning equal weights to all training samples.
- Training Weak Learners:
    - At each iteration, a Naive Bayes classifier is trained on a weighted sample of the training data. Weighted sampling ensures that misclassified instances in previous iterations are given higher priority.
    - The trained classifier calculates the class probabilities and predicts the outcomes for the full dataset.
- Error Calculation and Model Weighting:
    - The error rate of the model is computed based on its misclassification performance.
    - If the error rate exceeds a predefined threshold (e.g., 0.5), the iteration is halted as the model would not contribute positively to the ensemble.
    - A weight is assigned to the model based on its accuracy. Models with lower error rates receive higher weights.
- Updating Sample Weights:
    - The sample weights are adjusted, increasing for misclassified samples. This ensures that subsequent iterations focus more on challenging cases.
    - The weights are normalized to maintain a valid probability distribution.
- Final Prediction:
    - During prediction, the weighted outputs of all Naive Bayes models are aggregated. The final classification is determined by the majority vote or the highest combined probability.

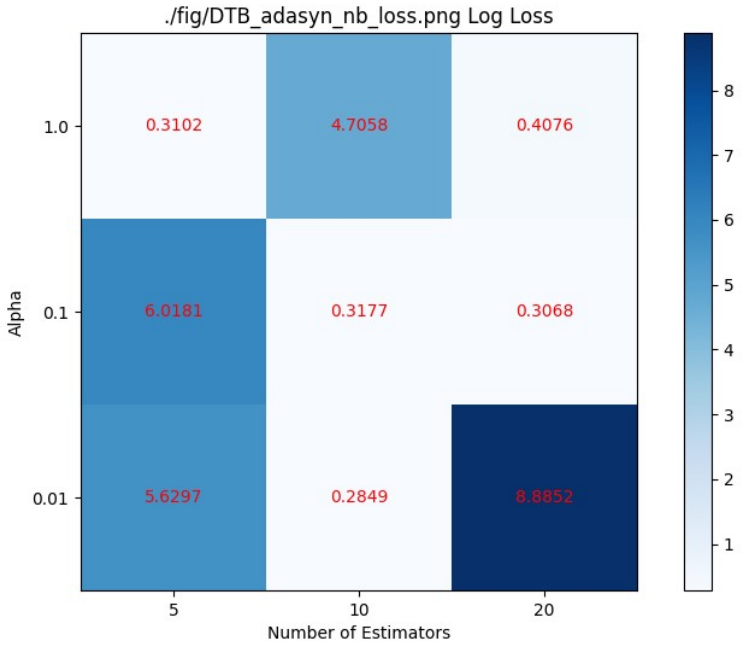
Advantages

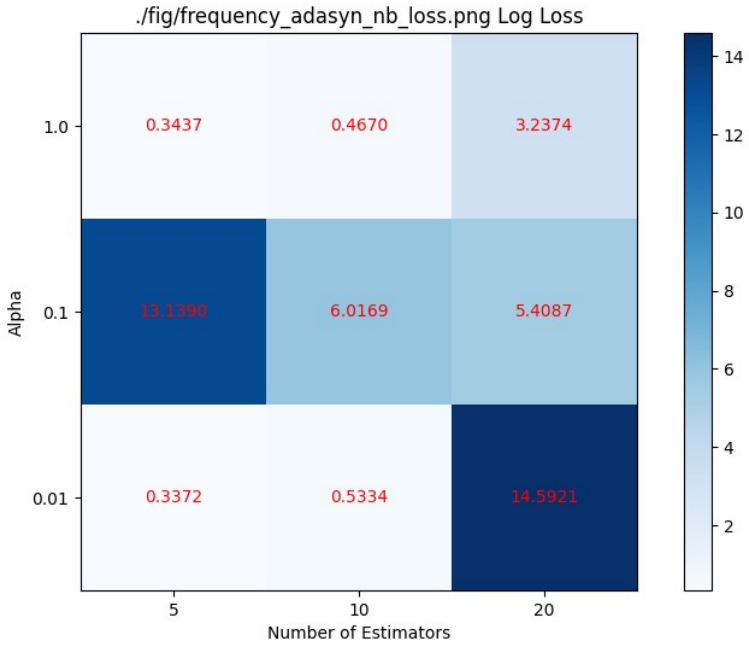
- **Improved Accuracy:** By combining multiple Naive Bayes classifiers, the ensemble reduces the bias and variance associated with individual models.
- **Adaptability:** The iterative weighting mechanism allows the ensemble to adapt to challenging data points, improving performance on complex datasets.

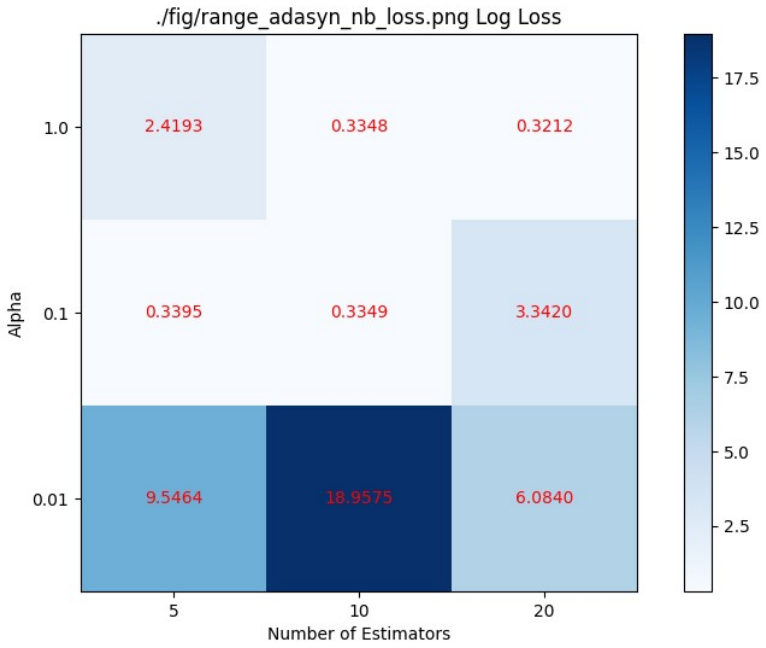
Limitations

- **Computational Complexity:** Training multiple models and updating weights across iterations increase the computational cost compared to a standalone Naive Bayes classifier.
- **Sensitivity to Outliers:** Adaboost can be sensitive to noisy data or outliers, as misclassified samples receive higher weights, potentially amplifying their impact.

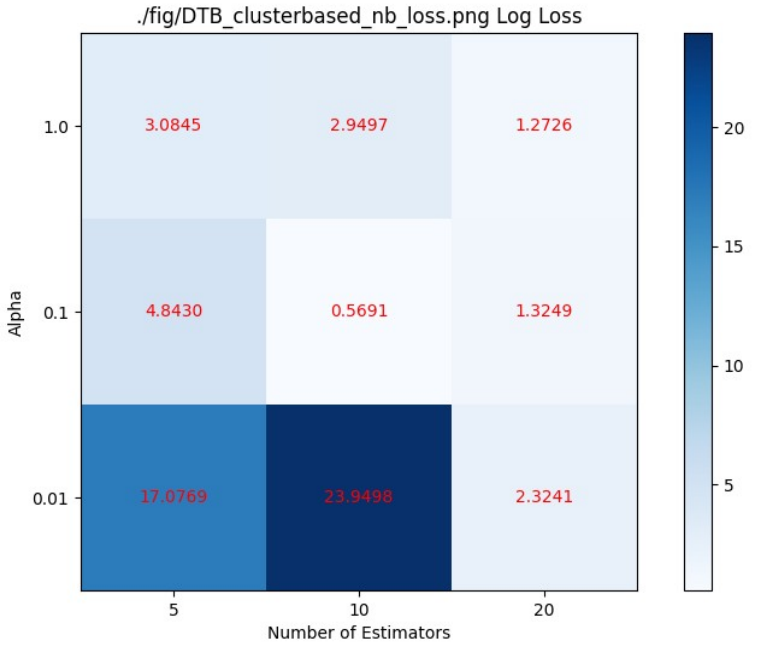
Adasyn:

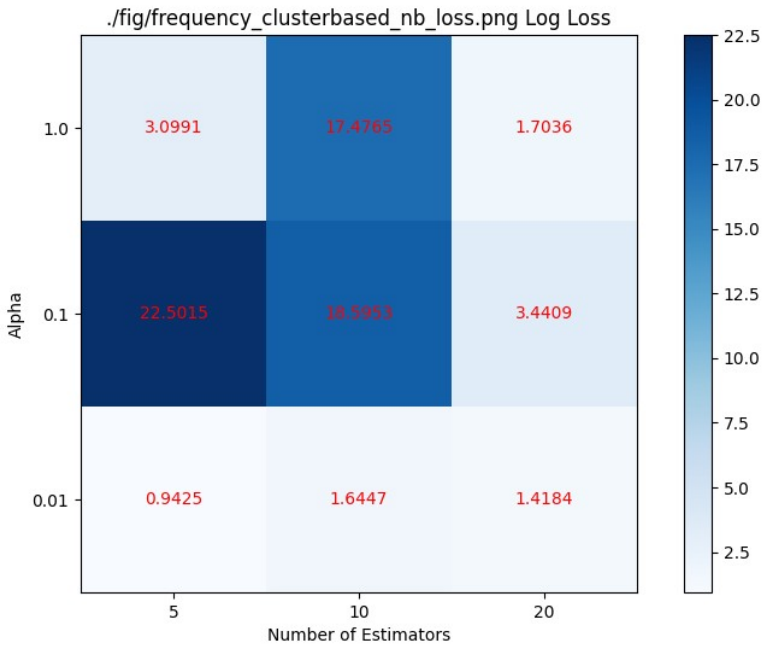

./fig/DTB_adasyn_nb_loss.png Log Loss
0.3102
4.7058
0.4076
6.0181
0.3177
0.3068
5.6297
0.2849
8.8852
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
8
7
6
5
4
3
2
1

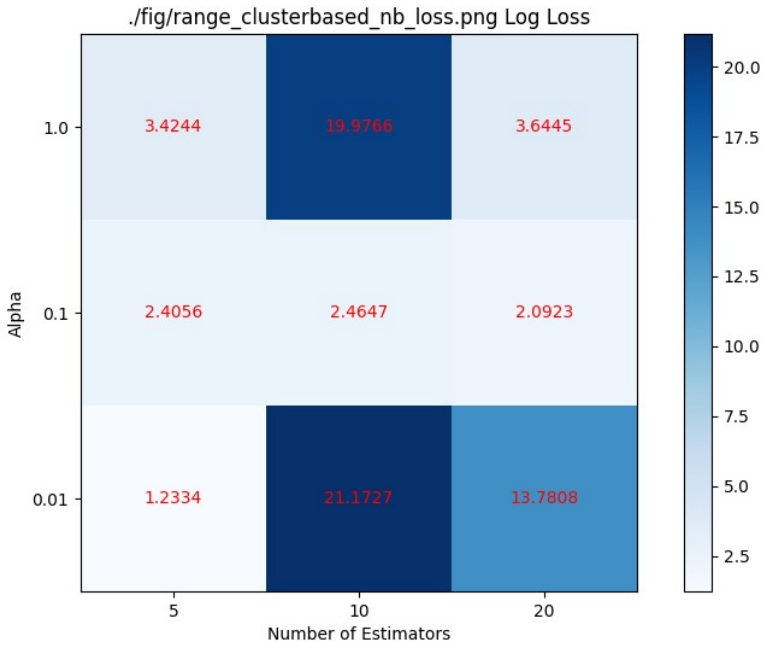

./fig/frequency_adasyn_nb_loss.png Log Loss
0.3437
0.4670
3.2374
13.1390
6.0169
5.4087
0.3372
0.5334
14.5921
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
14
12
10
8
6
4
2


./fig/range_adasyn_nb_loss.png Log Loss
Alpha
1.0
0.1
0.01
2.4193
0.3348
0.3212
0.3395
0.3349
3.3420
9.5464
18.9575
6.0840
5
10
20
Number of Estimators
17.5
15.0
12.5
10.0
7.5
5.0
2.5

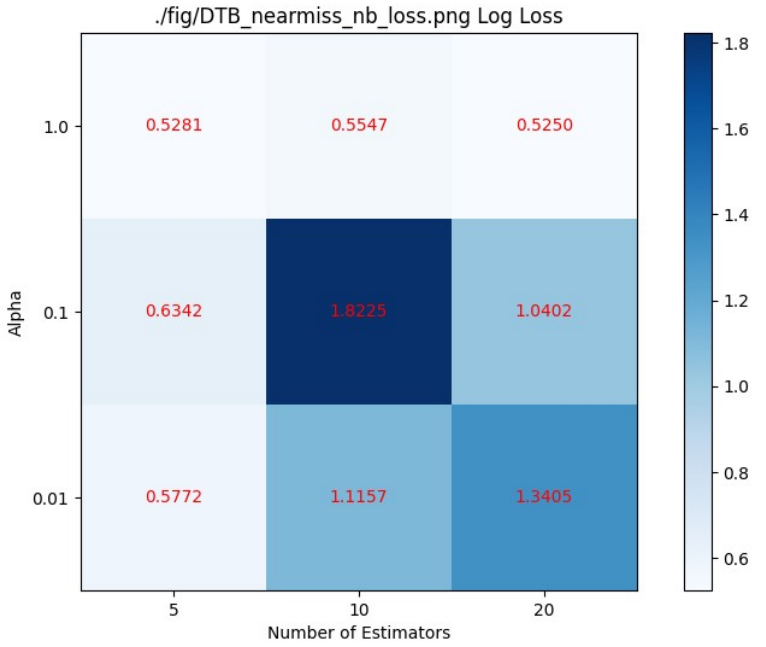
Clusterbased:

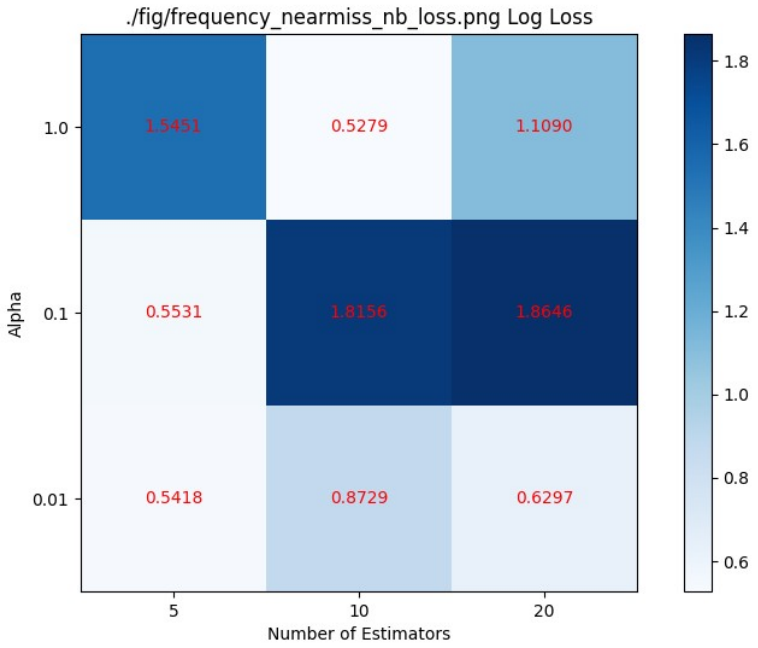


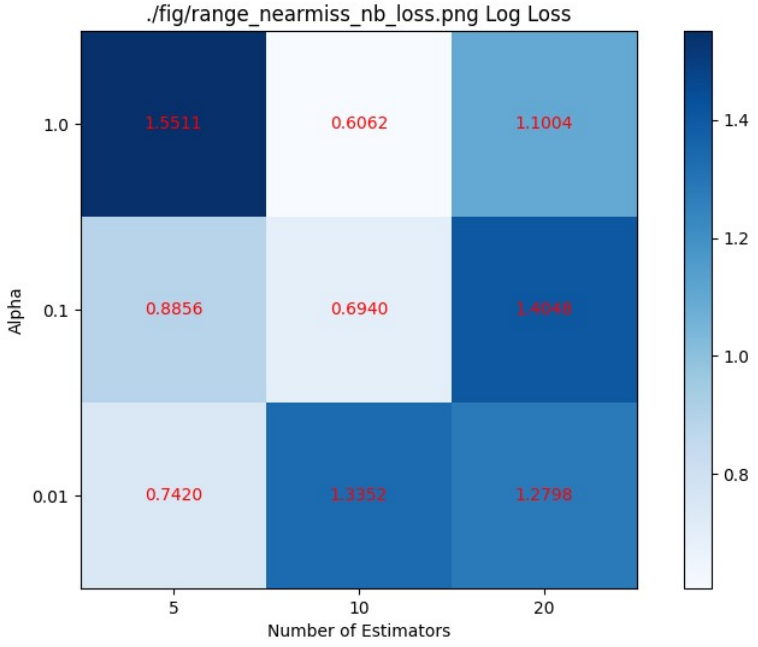



./fig/range_clusterbased_nb_loss.png Log Loss
Alpha
1.0
0.1
0.01
3.4244
19.9766
3.6445
2.4056
2.4647
2.0923
1.2334
21.1727
13.7808
5
10
20
Number of Estimators
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5

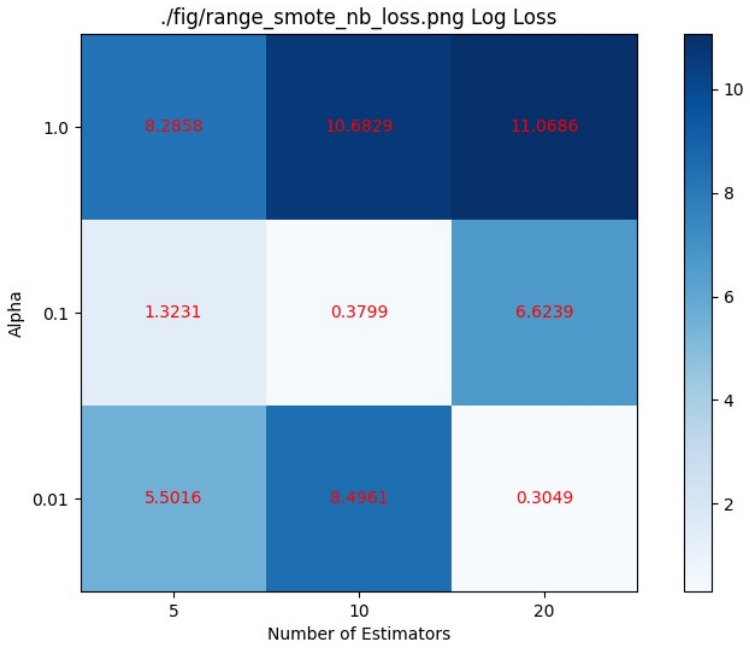
Nearmiss:

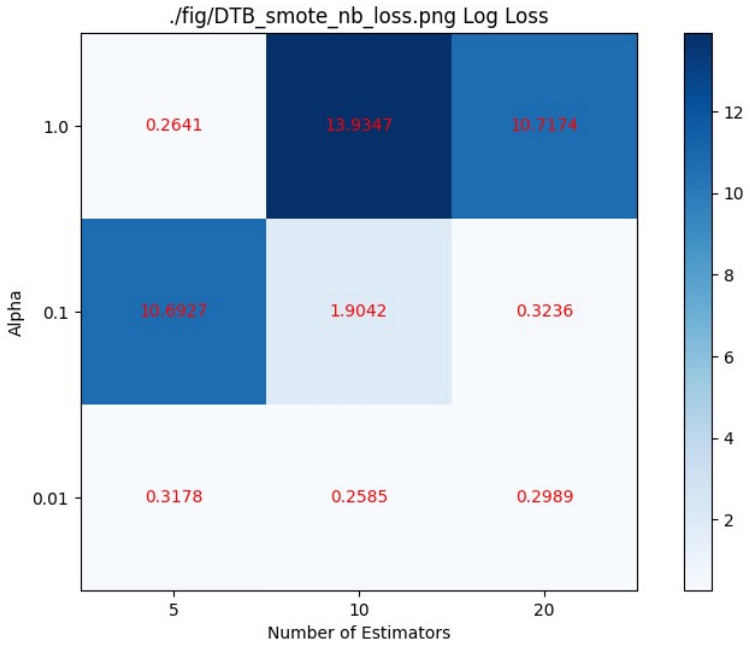

./fig/DTB_nearmiss_nb_loss.png Log Loss
Alpha
1.0
0.1
0.01
0.5281
0.5547
0.5250
0.6342
1.8225
1.0402
0.5772
1.1157
1.3405
5
10
20
Number of Estimators
1.8
1.6
1.4
1.2
1.0
0.8
0.6

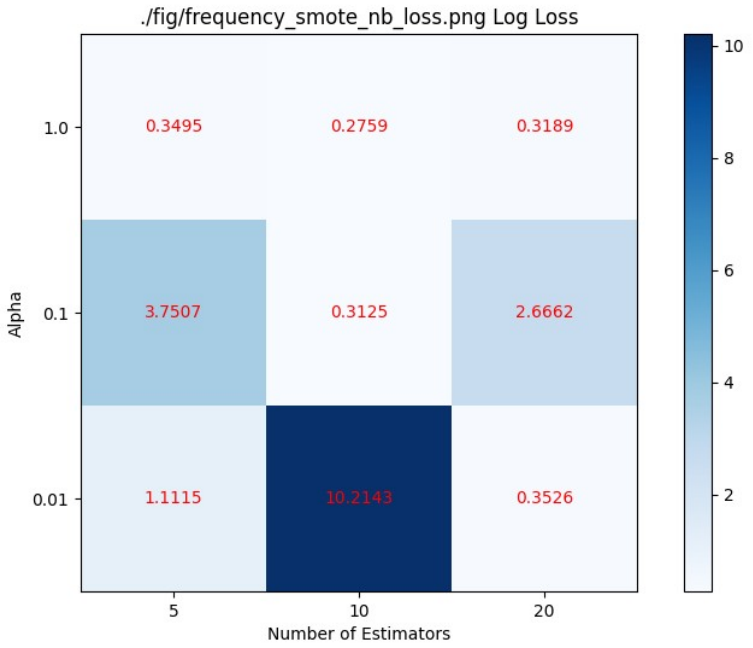

./fig/frequency_nearmiss_nb_loss.png Log Loss
1.5451
0.5279
1.1090
0.5531
1.8156
1.8646
0.5418
0.8729
0.6297
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
1.8
1.6
1.4
1.2
1.0
0.8
0.6


./fig/range_nearmiss_nb_loss.png Log Loss
1.5511
0.6062
1.1004
0.8856
0.6940
1.4048
0.7420
1.3352
1.2798
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
1.4
1.2
1.0
0.8

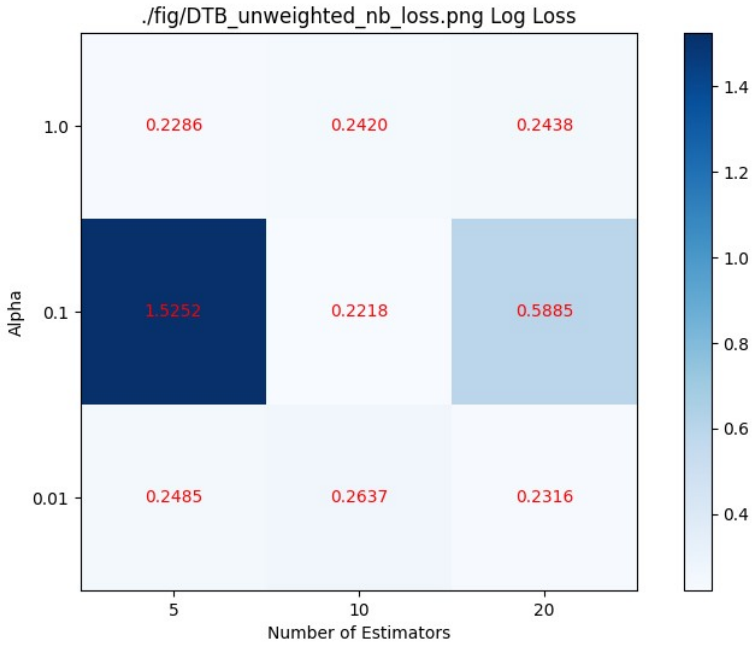
SMOTE:

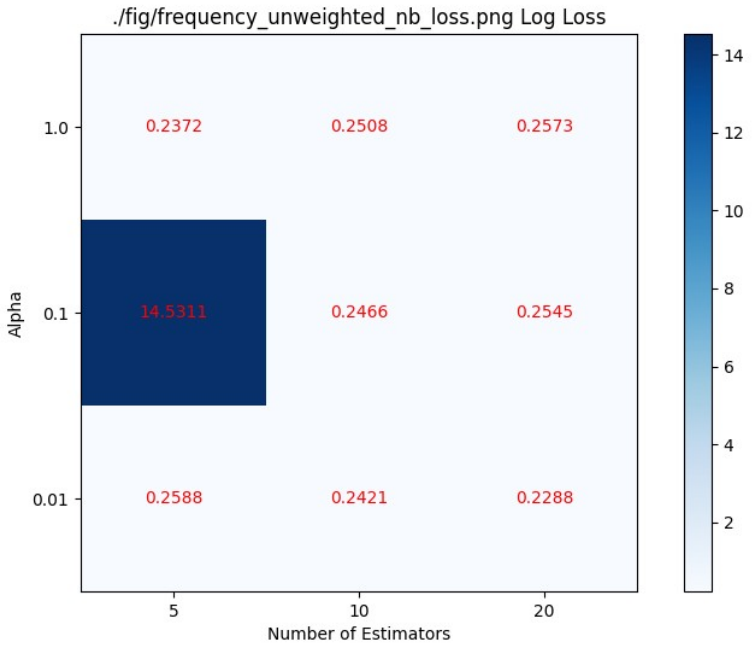

./fig/range_smote_nb_loss.png Log Loss
8.2858
10.6829
11.0686
1.3231
0.3799
6.6239
5.5016
8.4961
0.3049
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
10
8
6
4
2

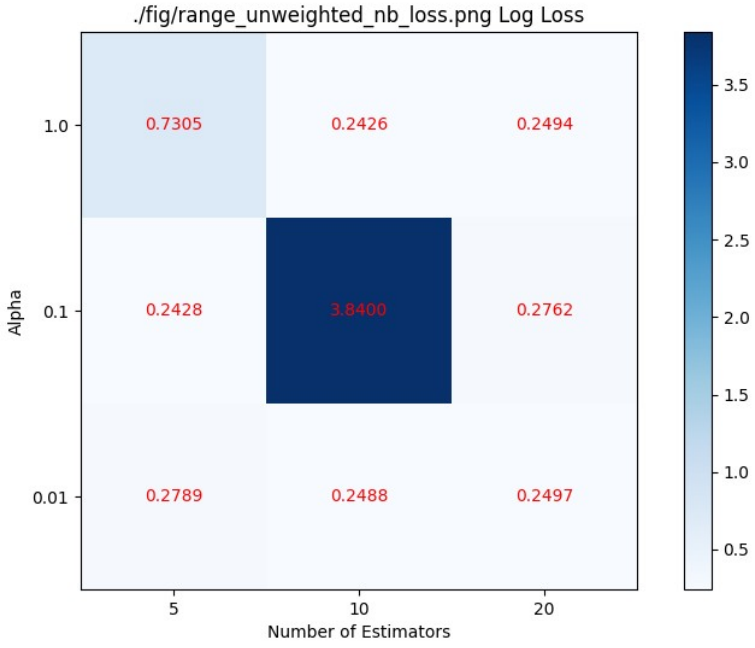

./fig/DTB_smote_nb_loss.png Log Loss
0.2641
13.9347
10.7174
10.6927
1.9042
0.3236
0.3178
0.2585
0.2989
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
12
10
8
6
4
2


./fig/frequency_smote_nb_loss.png Log Loss
0.3495
0.2759
0.3189
3.7507
0.3125
2.6662
1.1115
10.2143
0.3526
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
10
8
6
4
2

Unweighted:

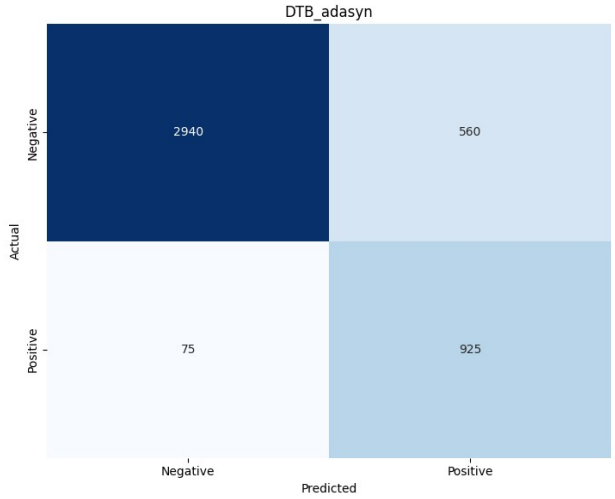

./fig/DTB_unweighted_nb_loss.png Log Loss
0.2286
0.2420
0.2438
1.5252
0.2218
0.5885
0.2485
0.2637
0.2316
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
1.4
1.2
1.0
0.8
0.6
0.4


./fig/frequency_unweighted_nb_loss.png Log Loss
0.2372
0.2508
0.2573
14.5311
0.2466
0.2545
0.2588
0.2421
0.2288
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
14
12
10
8
6
4
2


./fig/range_unweighted_nb_loss.png Log Loss
0.7305
0.2426
0.2494
0.2428
3.8400
0.2762
0.2789
0.2488
0.2497
1.0
0.1
0.01
Alpha
5
10
20
Number of Estimators
3.5
3.0
2.5
2.0
1.5
1.0
0.5

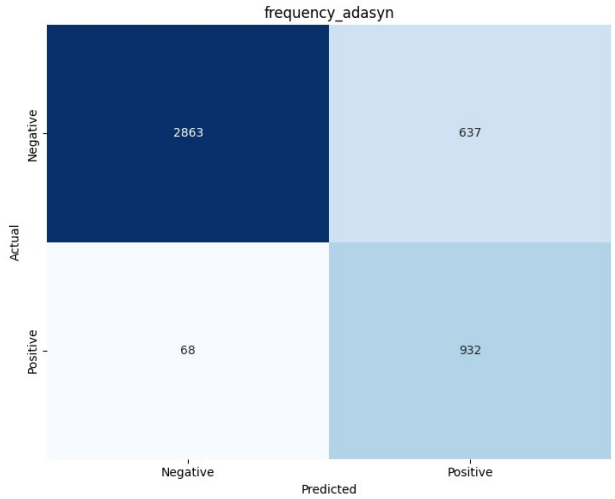
Confusion Matrices For Adaboost Naive Bayes Original Test and Validation Sets

Adasyn:



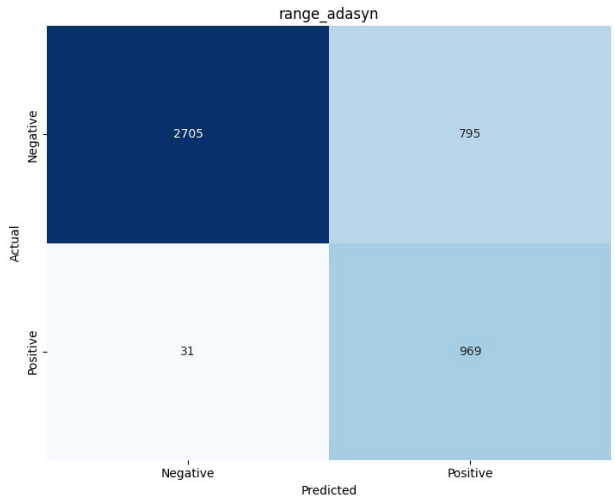
Metrics

Accuracy: 0.86
Precision: 0.62
Recall: 0.93
Specificity: 0.84
F1 Score: 0.74



Metrics

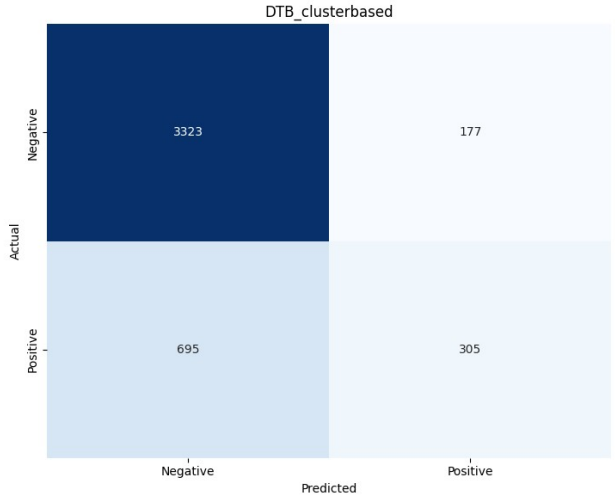
Accuracy: 0.84
Precision: 0.59
Recall: 0.93
Specificity: 0.82
F1 Score: 0.73



Metrics

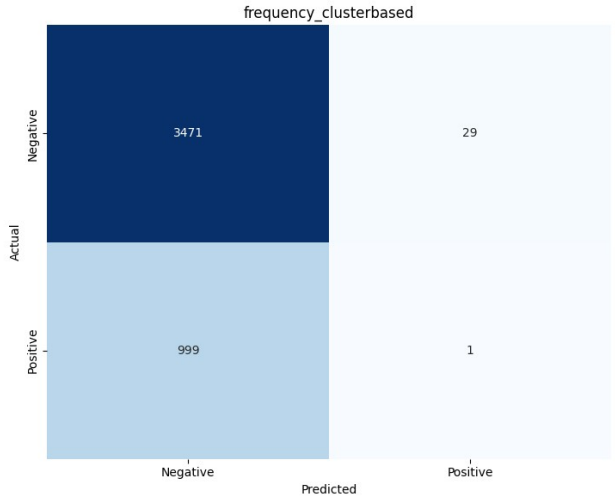
Accuracy: 0.82
Precision: 0.55
Recall: 0.97
Specificity: 0.77
F1 Score: 0.70

Clusterbased:



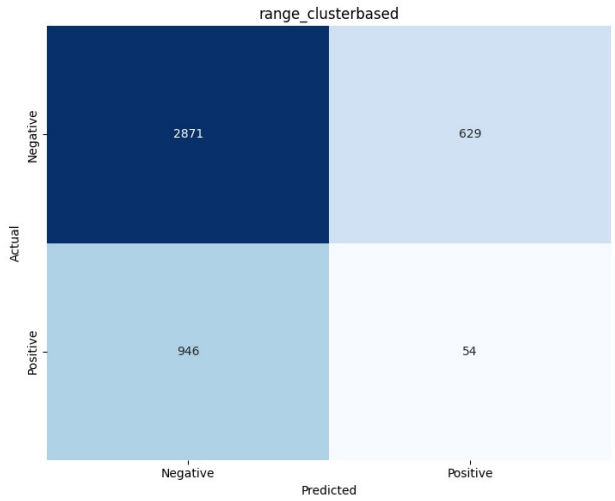
Metrics

Accuracy: 0.81
Precision: 0.63
Recall: 0.30
Specificity: 0.95
F1 Score: 0.41



Metrics

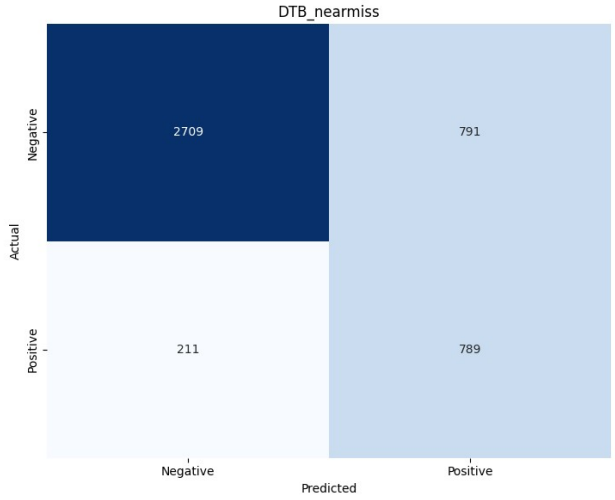
Accuracy: 0.77
Precision: 0.03
Recall: 0.00
Specificity: 0.99
F1 Score: 0.00



Metrics

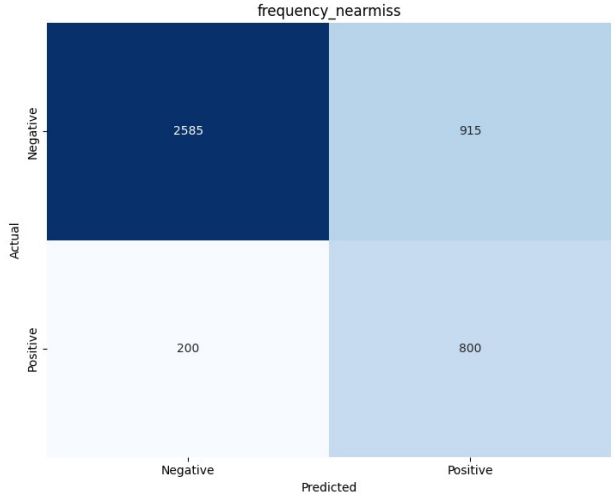
Accuracy: 0.65
Precision: 0.08
Recall: 0.05
Specificity: 0.82
F1 Score: 0.06

Nearmiss:



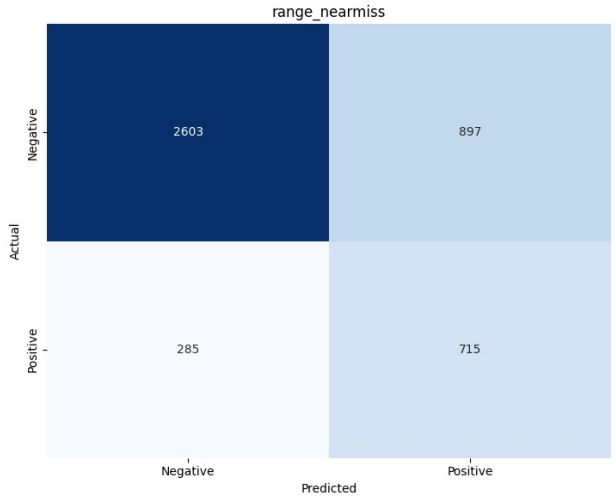
Metrics

Accuracy: 0.78
Precision: 0.50
Recall: 0.79
Specificity: 0.77
F1 Score: 0.61



Metrics

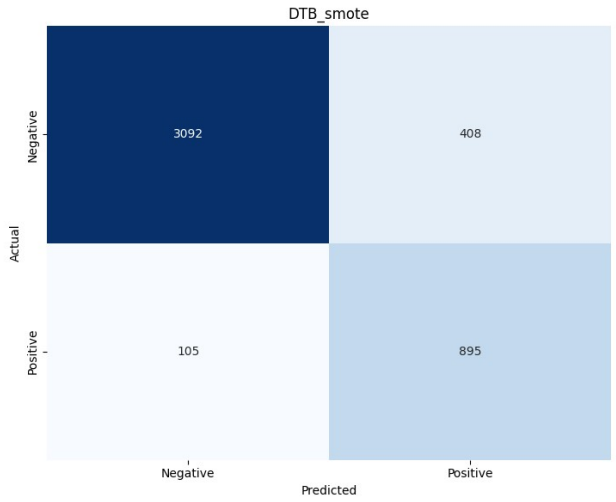
Accuracy: 0.75
Precision: 0.47
Recall: 0.80
Specificity: 0.74
F1 Score: 0.59



Metrics

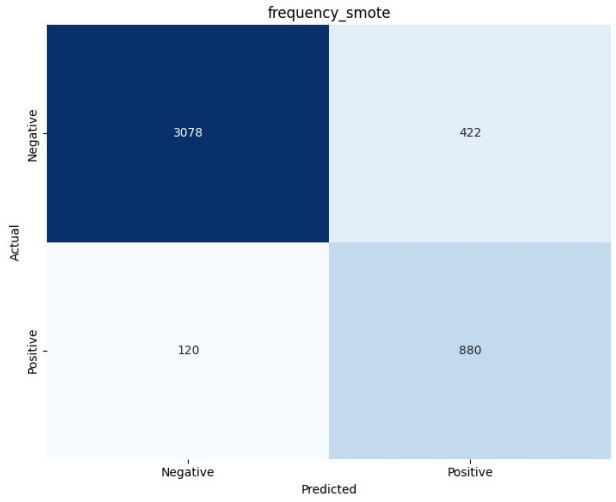
Accuracy: 0.74
Precision: 0.44
Recall: 0.71
Specificity: 0.74
F1 Score: 0.55

SMOTE:



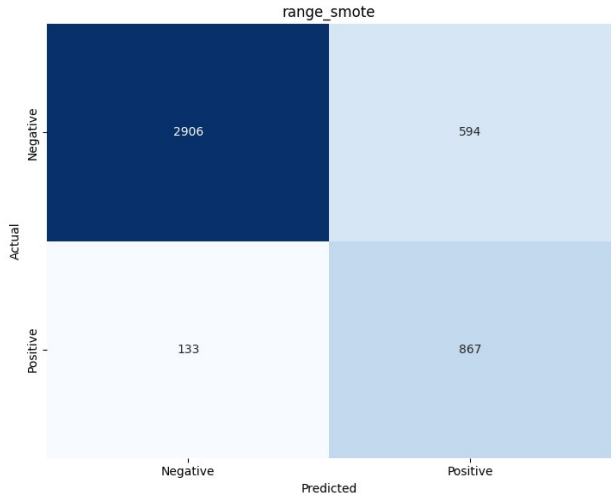
Metrics

Accuracy: 0.89
Precision: 0.69
Recall: 0.90
Specificity: 0.88
F1 Score: 0.78



Metrics

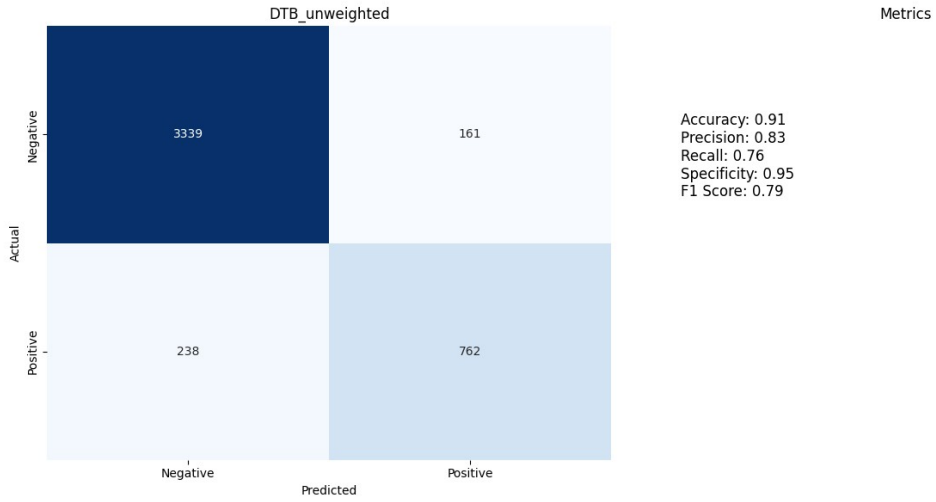
Accuracy: 0.88
Precision: 0.68
Recall: 0.88
Specificity: 0.88
F1 Score: 0.76

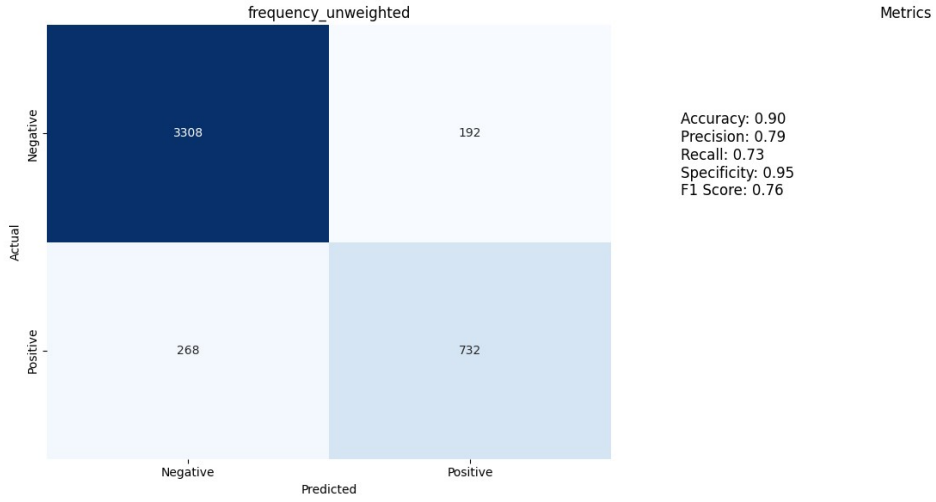


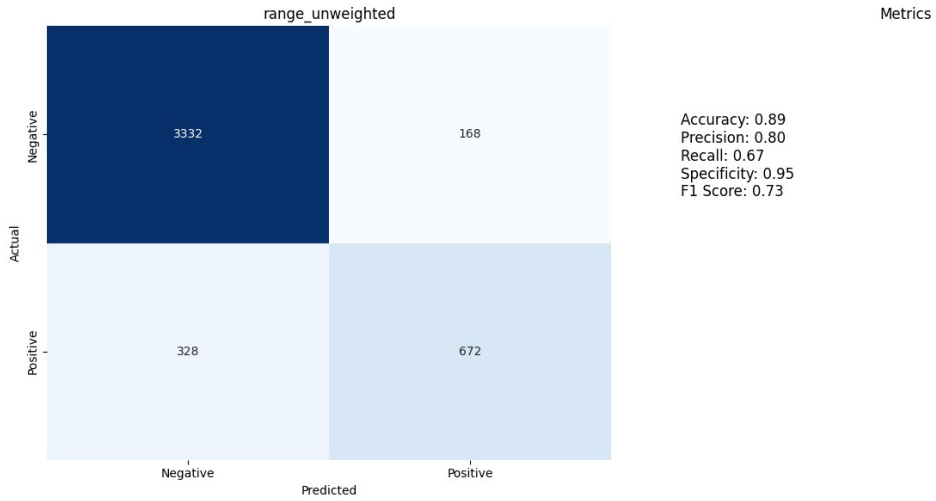
Metrics

Accuracy: 0.84
Precision: 0.59
Recall: 0.87
Specificity: 0.83
F1 Score: 0.70

Unweighted:

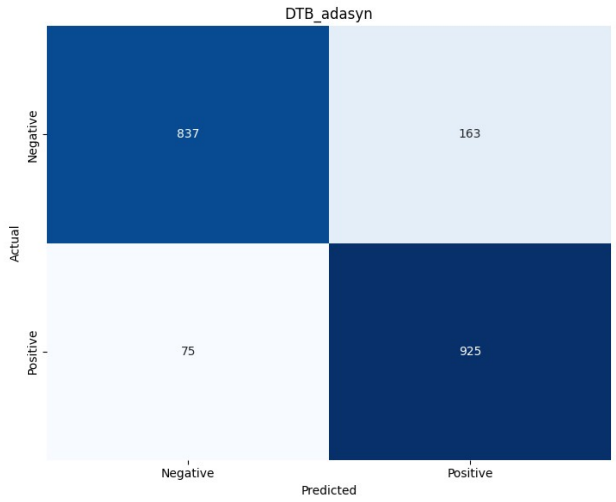






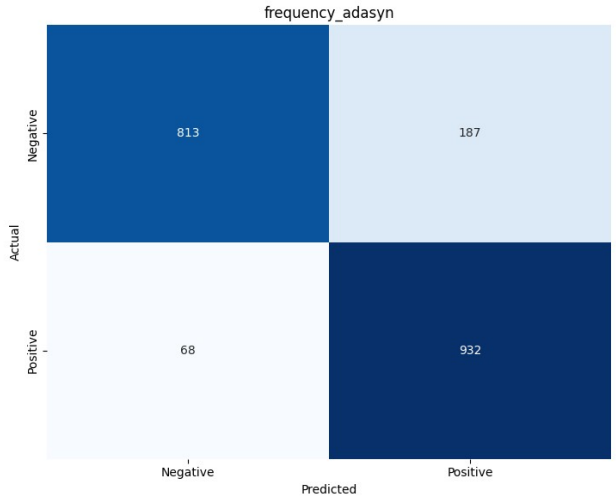
Undersampled Test and Validation Sets

Adasyn:



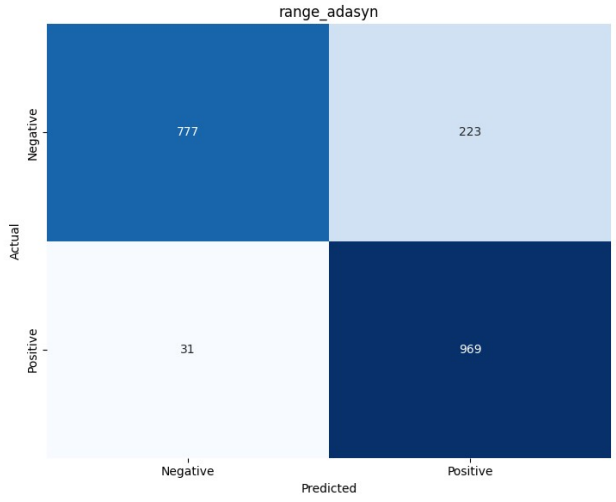
Metrics

Accuracy: 0.88
Precision: 0.85
Recall: 0.93
Specificity: 0.84
F1 Score: 0.89



Metrics

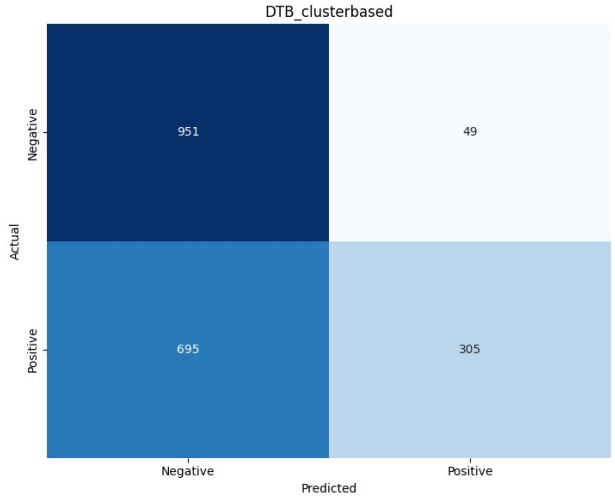
Accuracy: 0.87
Precision: 0.83
Recall: 0.93
Specificity: 0.81
F1 Score: 0.88



Metrics

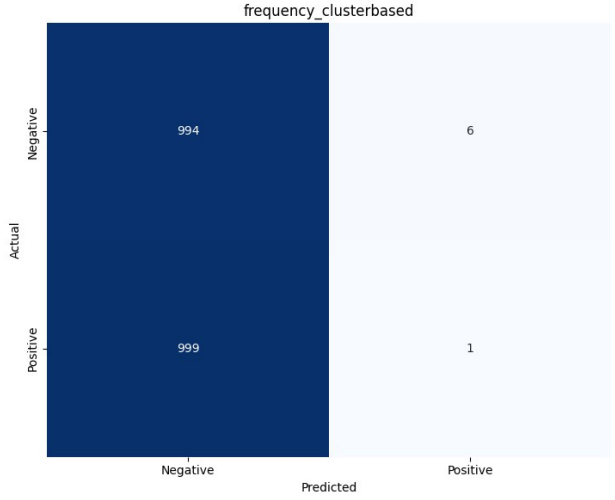
Accuracy: 0.87
Precision: 0.81
Recall: 0.97
Specificity: 0.78
F1 Score: 0.88

Clusterbased:



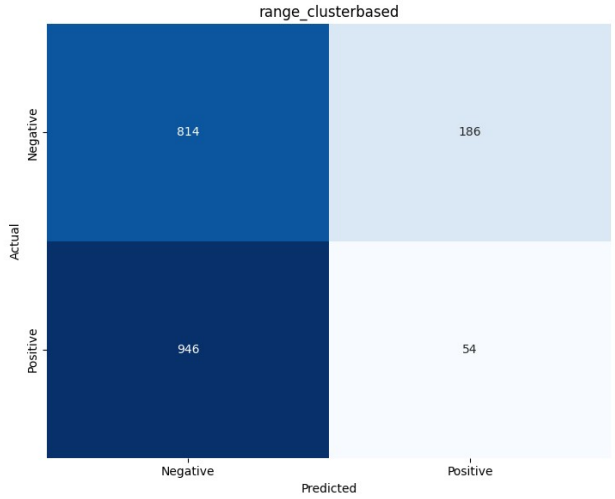
Metrics

Accuracy: 0.63
Precision: 0.86
Recall: 0.30
Specificity: 0.95
F1 Score: 0.45



Metrics

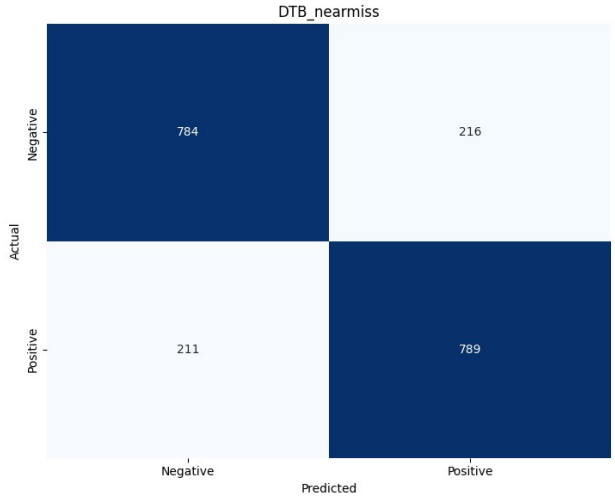
Accuracy: 0.50
Precision: 0.14
Recall: 0.00
Specificity: 0.99
F1 Score: 0.00



Metrics

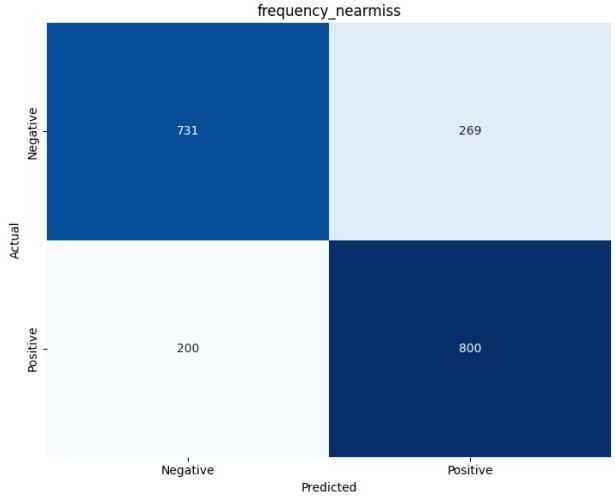
Accuracy: 0.43
Precision: 0.23
Recall: 0.05
Specificity: 0.81
F1 Score: 0.09

Nearmiss:



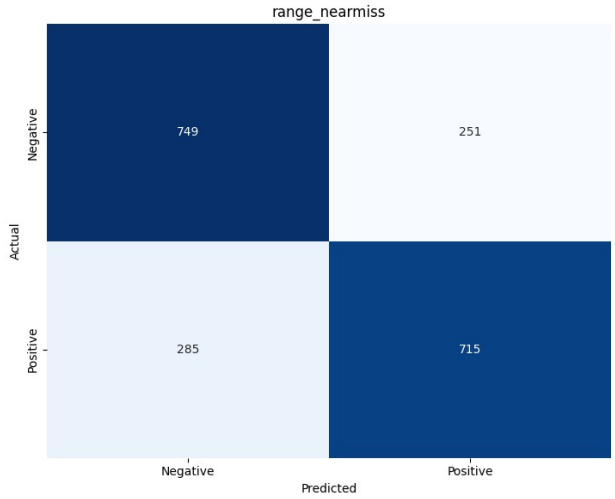
Metrics

Accuracy: 0.79
Precision: 0.79
Recall: 0.79
Specificity: 0.78
F1 Score: 0.79



Metrics

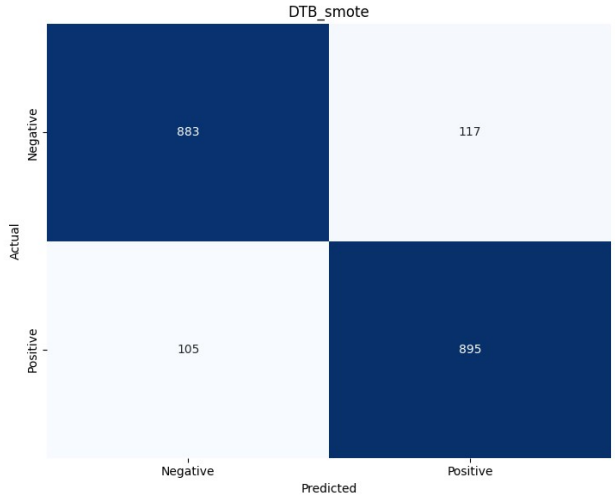
Accuracy: 0.77
Precision: 0.75
Recall: 0.80
Specificity: 0.73
F1 Score: 0.77

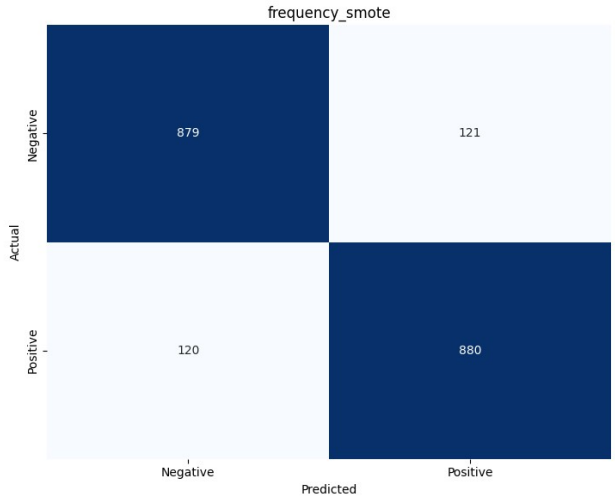


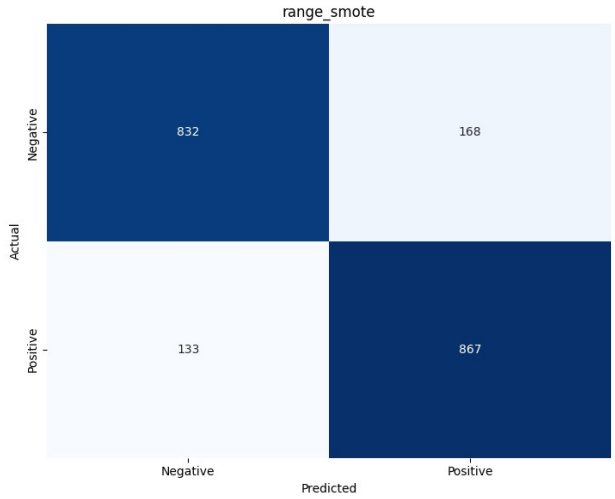
Metrics

Accuracy: 0.73
Precision: 0.74
Recall: 0.71
Specificity: 0.75
F1 Score: 0.73

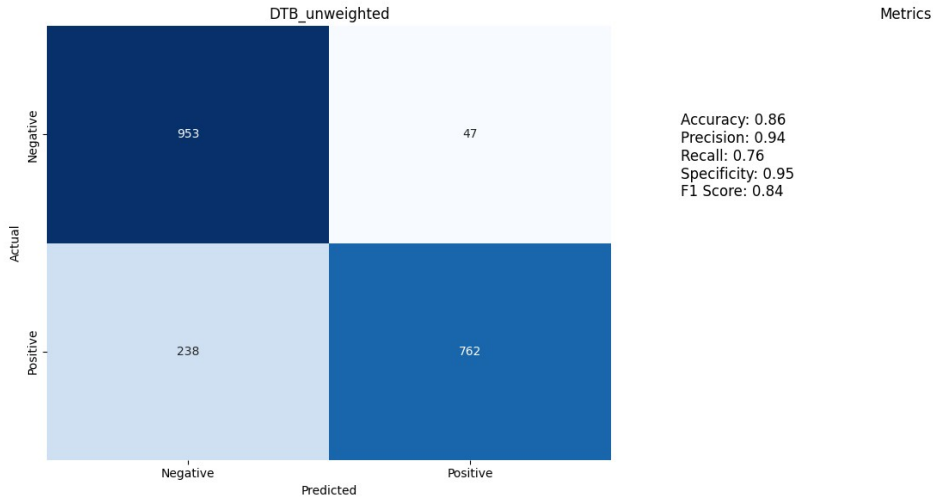
SMOTE:

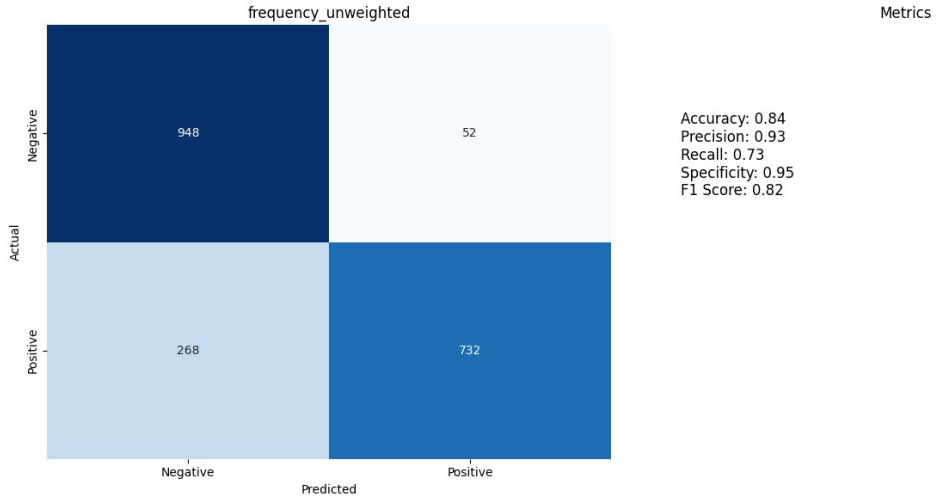

DTB_smote
Negative
Positive
883
117
105
895
Actual
Predicted
Metrics
Accuracy: 0.89
Precision: 0.88
Recall: 0.90
Specificity: 0.88
F1 Score: 0.89

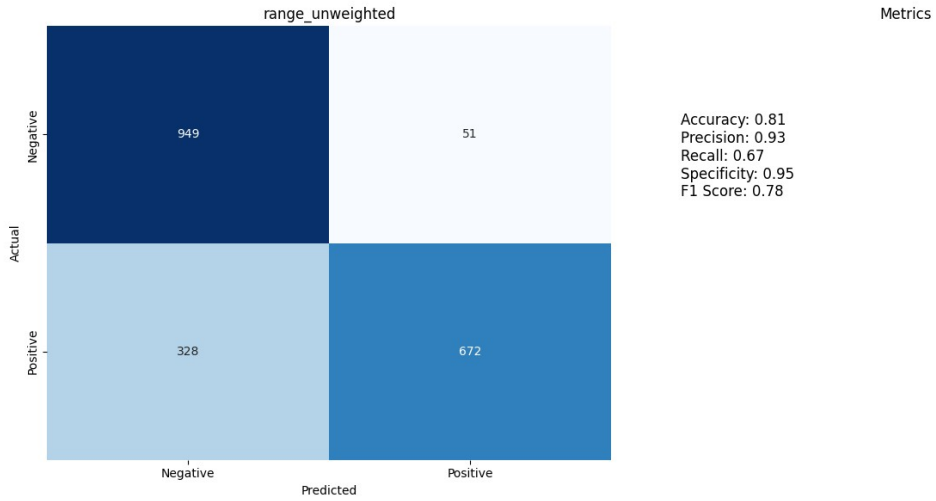

frequency_smote
Negative
Positive
879
121
120
880
Actual
Predicted
Metrics
Accuracy: 0.88
Precision: 0.88
Recall: 0.88
Specificity: 0.88
F1 Score: 0.88


range_smote
Negative
Positive
832
168
133
867
Actual
Predicted
Metrics
Accuracy: 0.85
Precision: 0.84
Recall: 0.87
Specificity: 0.83
F1 Score: 0.85

Unweighted:



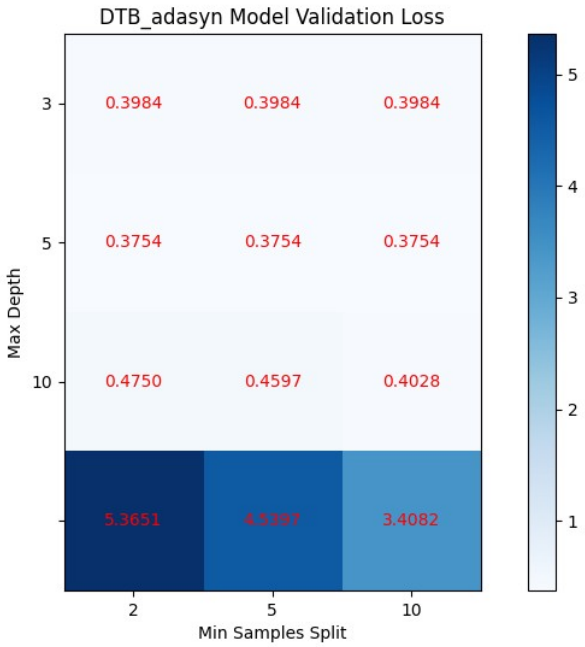


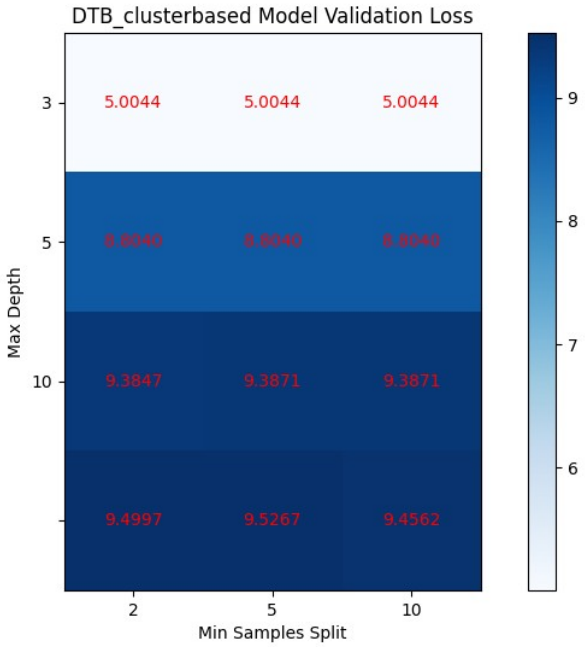


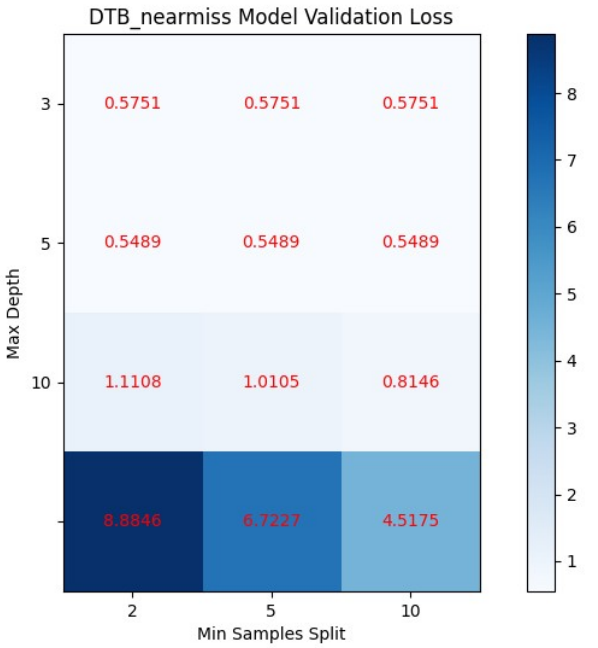
Decision tree

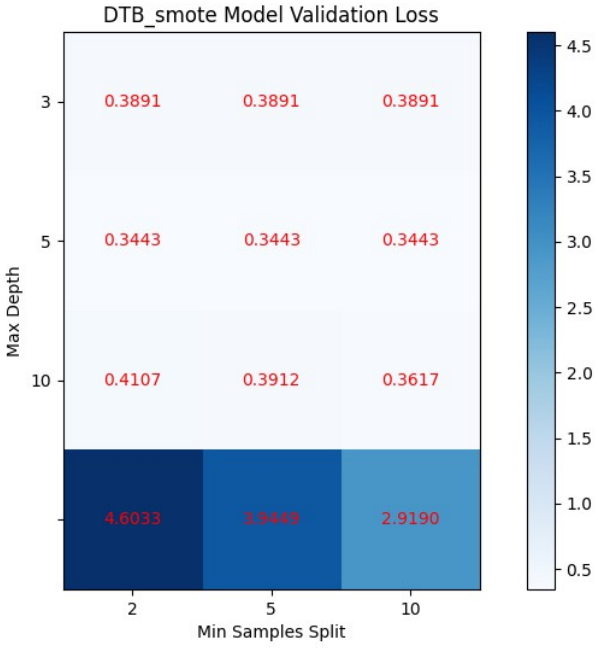
We used 3 different binning. Equal frequency, equal width and decision tree based binning and 5 different datasets Those datasets are created by applying ADASYN, SMOTE, cluster-based undersampling ,NearMiss. The fifth one is the original dataset's proper conversion and we used this set twice, applying weight and unweighted version. We want to compare

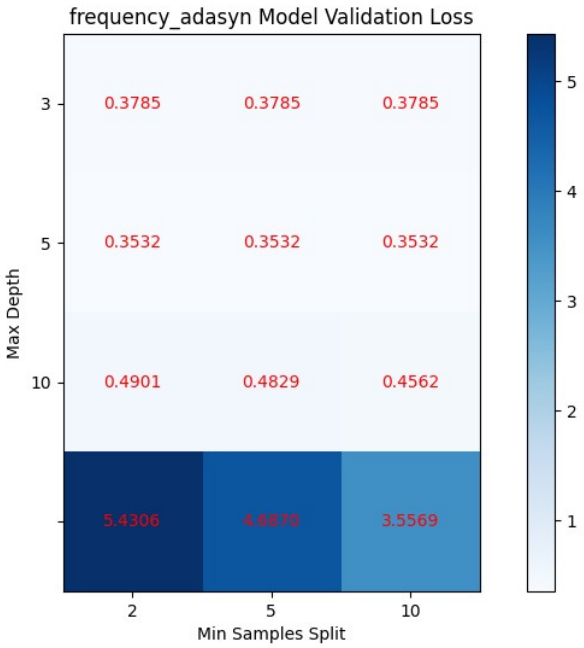
18 different models for each different dataset. We conducted 216 experiments, 12 for each model to get optimum hyperparameters. Following figures are the log loss matrix for these datasets.

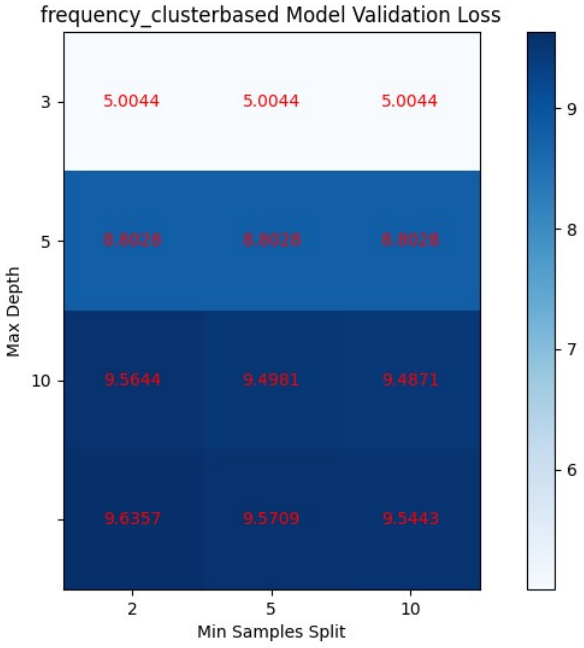


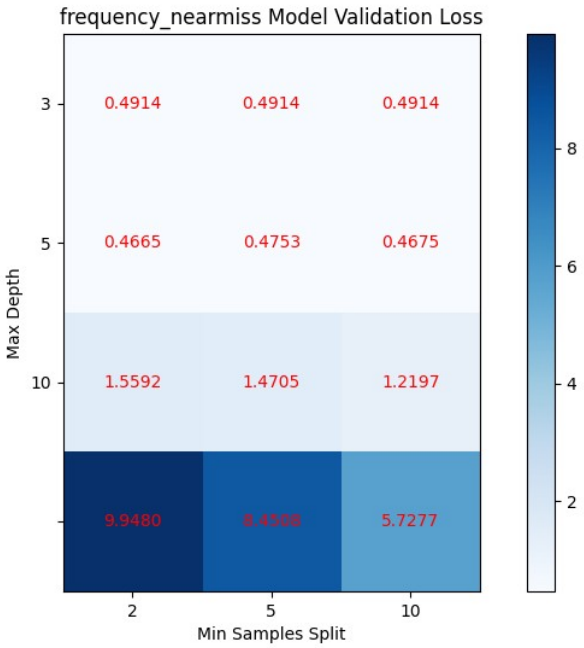


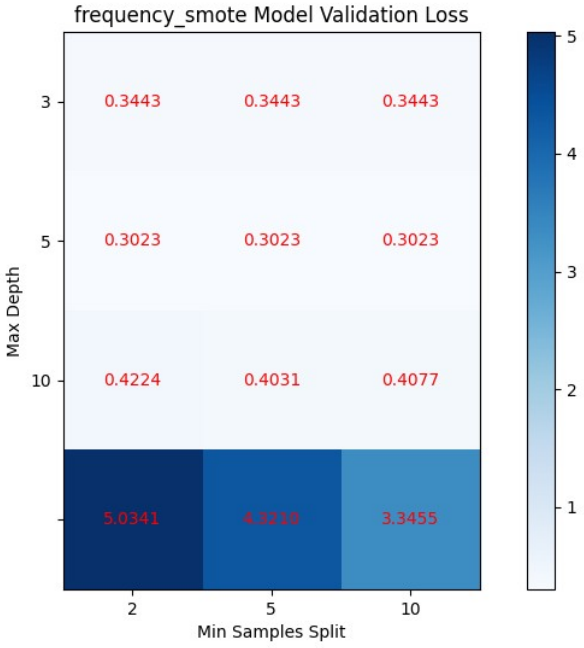

DTB_nearmiss Model Validation Loss
0.5751
0.5751
0.5751
0.5489
0.5489
0.5489
1.1108
1.0105
0.8146
8.8846
6.7227
4.5175
3
5
10
Max Depth
2
5
10
Min Samples Split
8
7
6
5
4
3
2
1

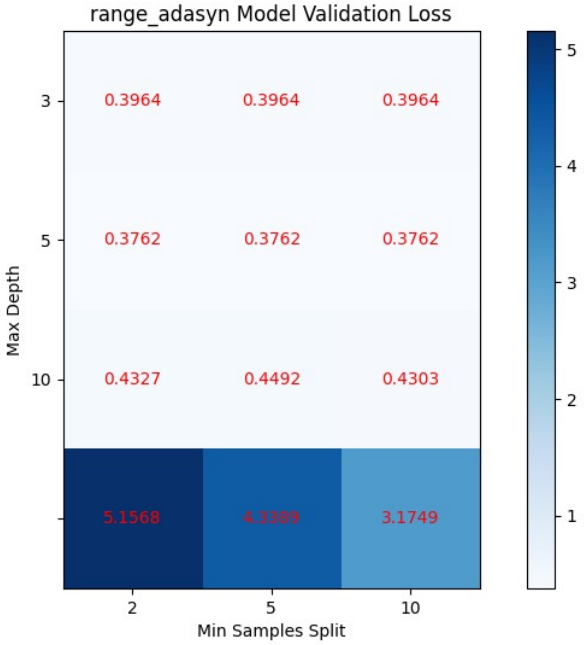

DTB_smote Model Validation Loss
0.3891
0.3891
0.3891
0.3443
0.3443
0.3443
0.4107
0.3912
0.3617
4.6033
3.9449
2.9190
3
5
10
Max Depth
2
5
10
Min Samples Split
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5

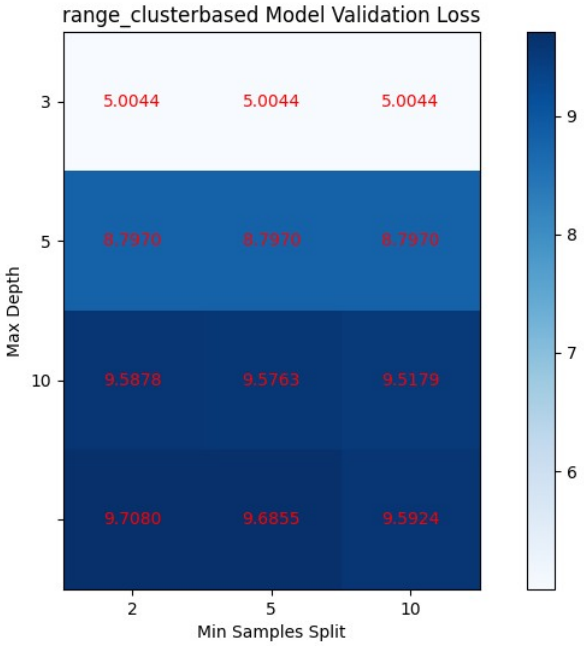

frequency_adasyn Model Validation Loss
0.3785
0.3785
0.3785
0.3532
0.3532
0.3532
0.4901
0.4829
0.4562
5.4306
4.6870
3.5569
3
5
10
Max Depth
2
5
10
Min Samples Split
5
4
3
2
1

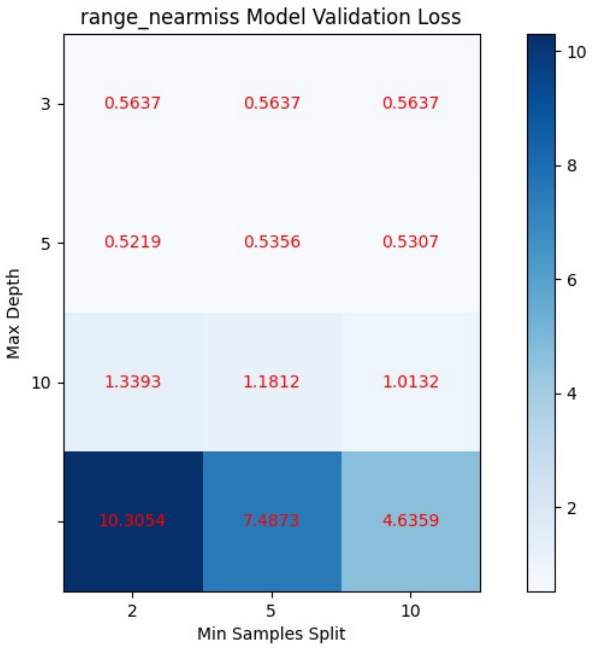

frequency_clusterbased Model Validation Loss
5.0044
5.0044
5.0044
8.8028
8.8028
8.8028
9.5644
9.4981
9.4871
9.6357
9.5709
9.5443
3
5
10
Max Depth
2
5
10
Min Samples Split
9
8
7
6

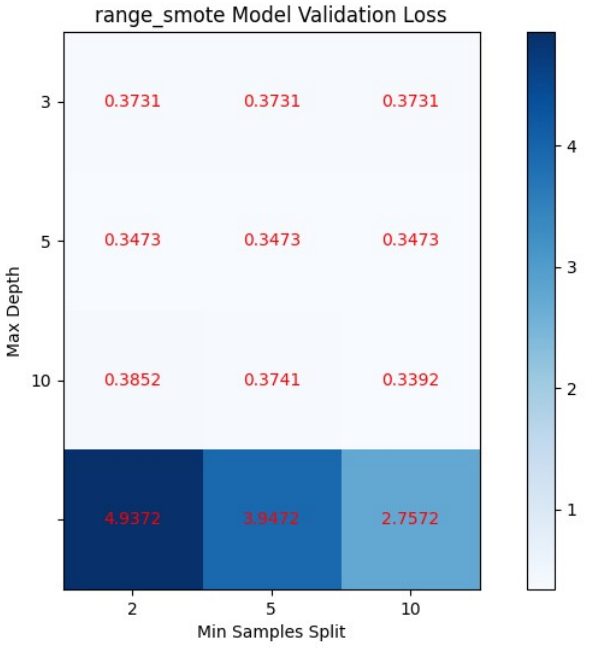

frequency_nearmiss Model Validation Loss
0.4914
0.4914
0.4914
0.4665
0.4753
0.4675
1.5592
1.4705
1.2197
9.9480
8.4508
5.7277
Max Depth
3
5
10
Min Samples Split
2
5
10
8
6
4
2

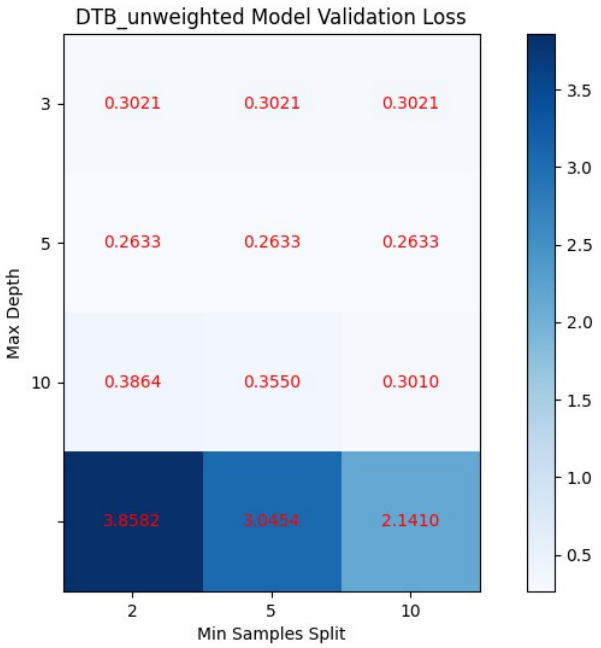

frequency_smote Model Validation Loss
0.3443
0.3443
0.3443
0.3023
0.3023
0.3023
0.4224
0.4031
0.4077
5.0341
4.3210
3.3455
Max Depth
3
5
10
Min Samples Split
2
5
10
5
4
3
2
1

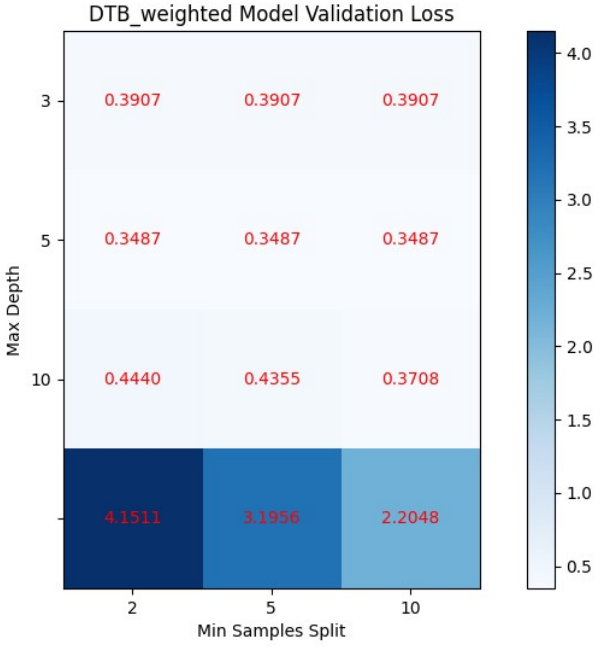

range_adasyn Model Validation Loss
0.3964
0.3964
0.3964
0.3762
0.3762
0.3762
0.4327
0.4492
0.4303
5.1568
4.3389
3.1749
3
5
10
Max Depth
2
5
10
Min Samples Split
5
4
3
2
1

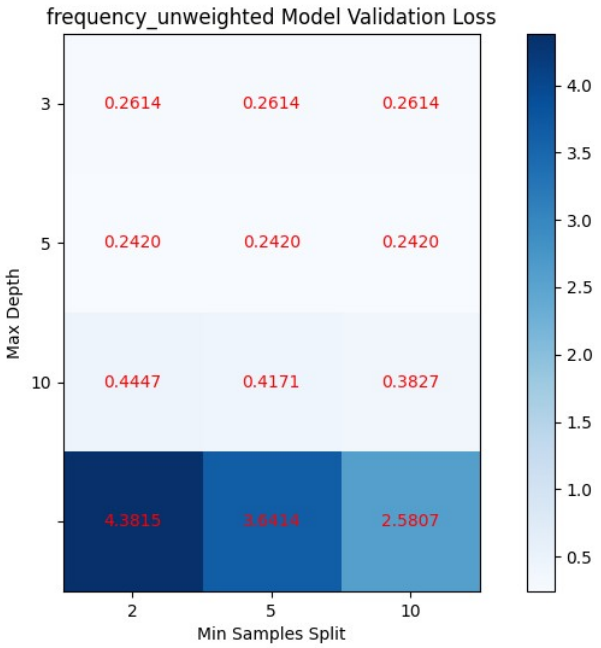

range_clusterbased Model Validation Loss
5.0044
5.0044
5.0044
8.7970
8.7970
8.7970
9.5878
9.5763
9.5179
9.7080
9.6855
9.5924
3
5
10
Max Depth
2
5
10
Min Samples Split
9
8
7
6

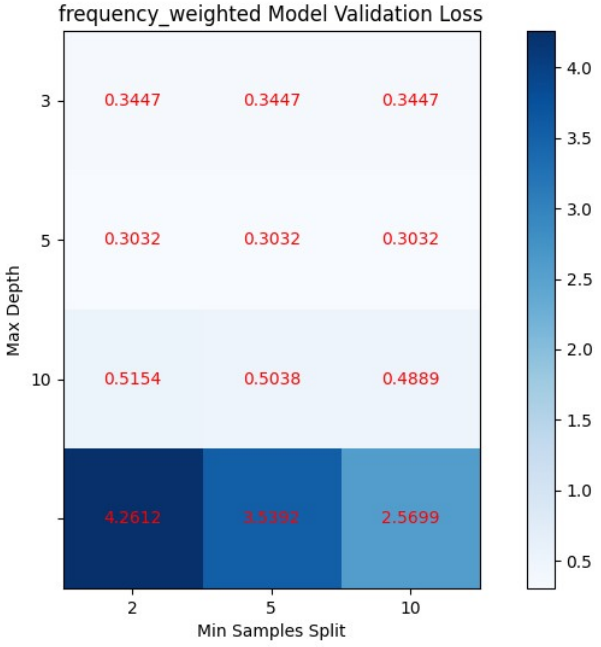

range_nearmiss Model Validation Loss
0.5637
0.5637
0.5637
0.5219
0.5356
0.5307
1.3393
1.1812
1.0132
10.3054
7.4873
4.6359
3
5
10
Max Depth
2
5
10
Min Samples Split
10
8
6
4
2

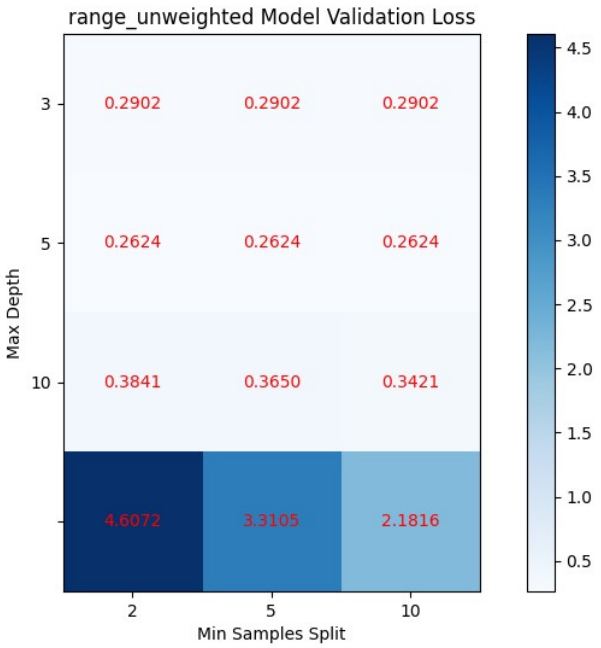

range_smote Model Validation Loss
0.3731
0.3731
0.3731
0.3473
0.3473
0.3473
0.3852
0.3741
0.3392
4.9372
3.9472
2.7572
3
5
10
Max Depth
2
5
10
Min Samples Split
4
3
2
1

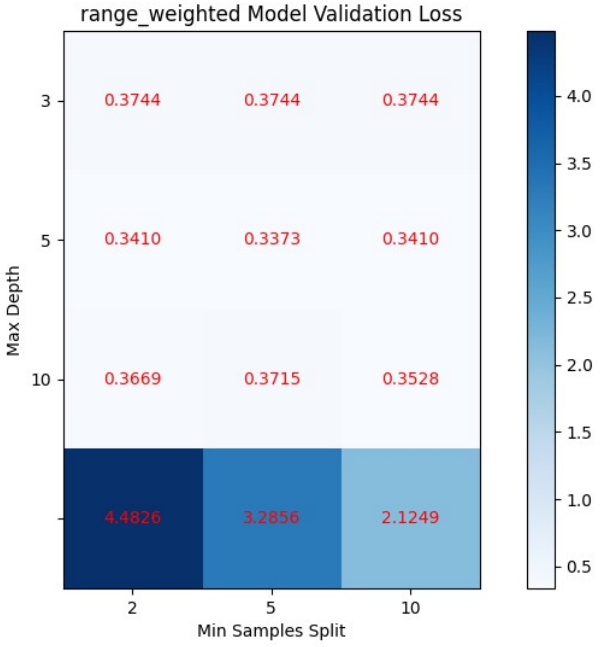

DTB_unweighted Model Validation Loss
Max Depth
3
5
10
0.3021
0.3021
0.3021
0.2633
0.2633
0.2633
0.3864
0.3550
0.3010
3.8582
3.0454
2.1410
2
5
10
Min Samples Split
3.5
3.0
2.5
2.0
1.5
1.0
0.5


DTB_weighted Model Validation Loss
Max Depth
3
5
10
0.3907
0.3907
0.3907
0.3487
0.3487
0.3487
0.4440
0.4355
0.3708
4.1511
3.1956
2.2048
2
5
10
Min Samples Split
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5

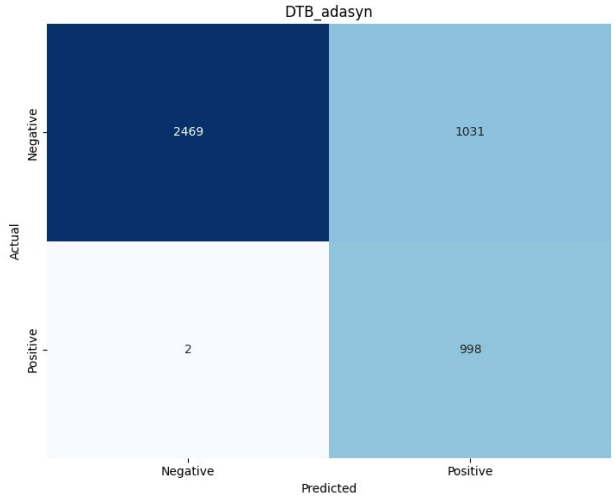

frequency_unweighted Model Validation Loss
Max Depth
3
5
10
0.2614
0.2614
0.2614
0.2420
0.2420
0.2420
0.4447
0.4171
0.3827
4.3815
3.6414
2.5807
2
5
10
Min Samples Split
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5


frequency_weighted Model Validation Loss
Max Depth
3
5
10
0.3447
0.3447
0.3447
0.3032
0.3032
0.3032
0.5154
0.5038
0.4889
4.2612
3.5392
2.5699
2
5
10
Min Samples Split
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5


range_unweighted Model Validation Loss
0.2902
0.2902
0.2902
0.2624
0.2624
0.2624
0.3841
0.3650
0.3421
4.6072
3.3105
2.1816
3
5
10
Max Depth
2
5
10
Min Samples Split
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5

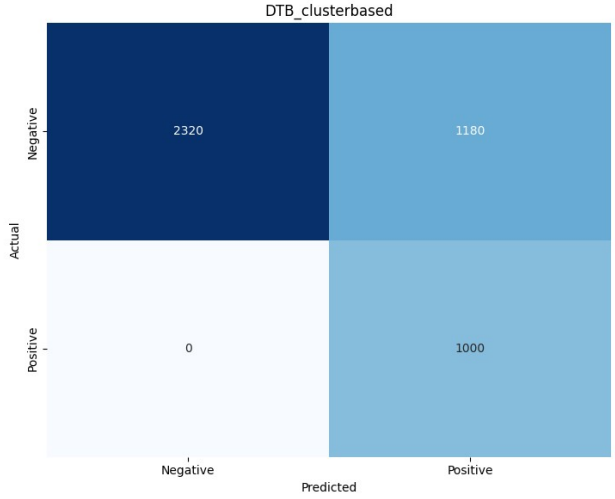

range_weighted Model Validation Loss
0.3744
0.3744
0.3744
0.3410
0.3373
0.3410
0.3669
0.3715
0.3528
4.4826
3.2856
2.1249
3
5
10
Max Depth
2
5
10
Min Samples Split
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5

Confusion matrix and metric results



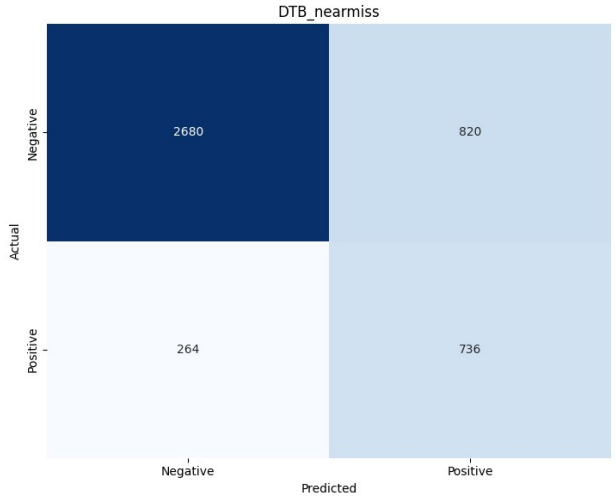
Metrics

Accuracy: 0.77
Precision: 0.49
Recall (Sensitivity): 1.00
Specificity: 0.71
F1 Score: 0.66



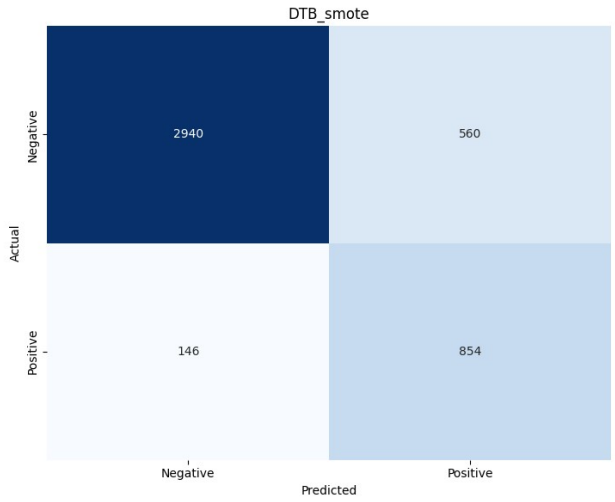
Metrics

Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63



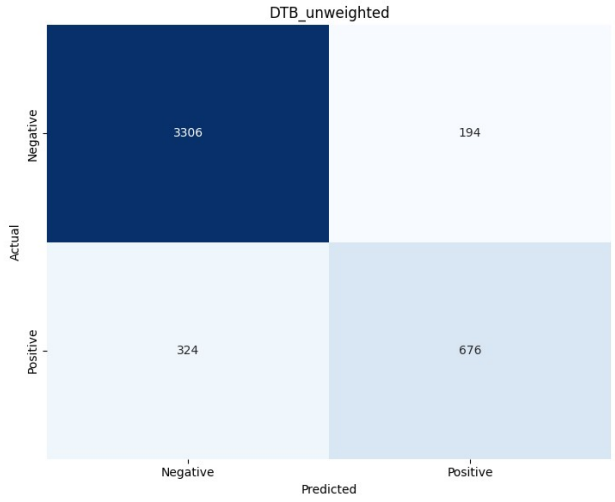
Metrics

Accuracy: 0.76
Precision: 0.47
Recall (Sensitivity): 0.74
Specificity: 0.77
F1 Score: 0.58



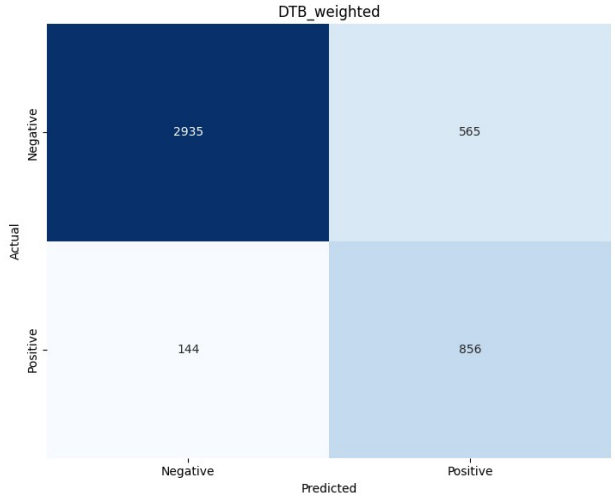
Metrics

Accuracy: 0.84
Precision: 0.60
Recall (Sensitivity): 0.85
Specificity: 0.84
F1 Score: 0.71



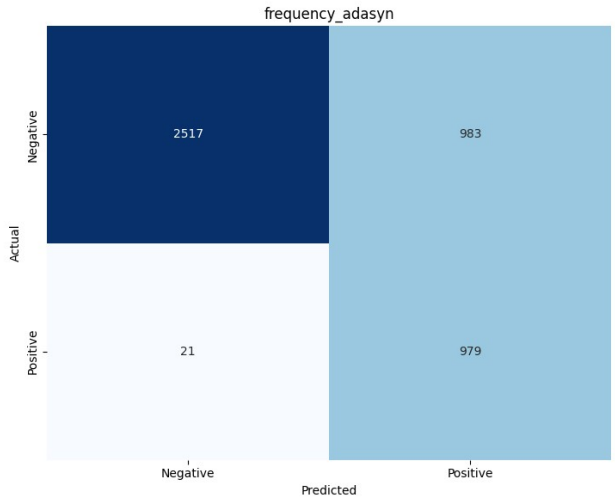
Metrics

Accuracy: 0.88
Precision: 0.78
Recall (Sensitivity): 0.68
Specificity: 0.94
F1 Score: 0.72



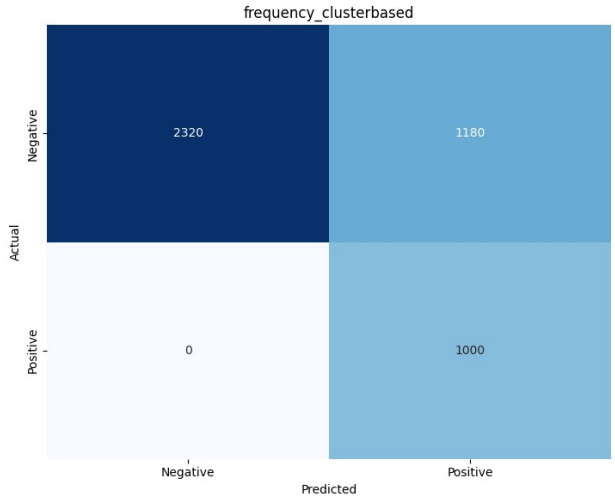
Metrics

Accuracy: 0.84
Precision: 0.60
Recall (Sensitivity): 0.86
Specificity: 0.84
F1 Score: 0.71



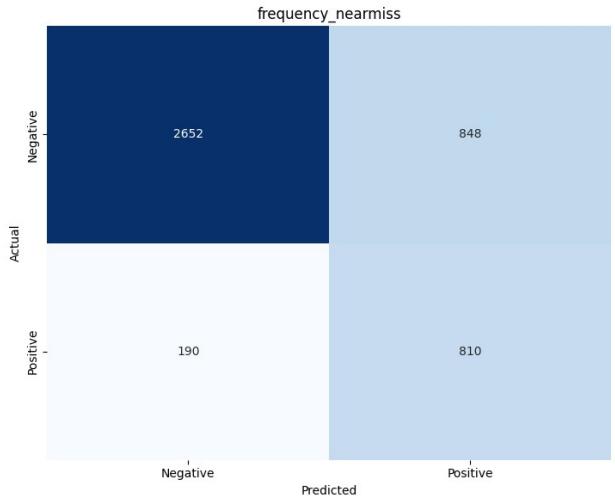
Metrics

Accuracy: 0.78
Precision: 0.50
Recall (Sensitivity): 0.98
Specificity: 0.72
F1 Score: 0.66



Metrics

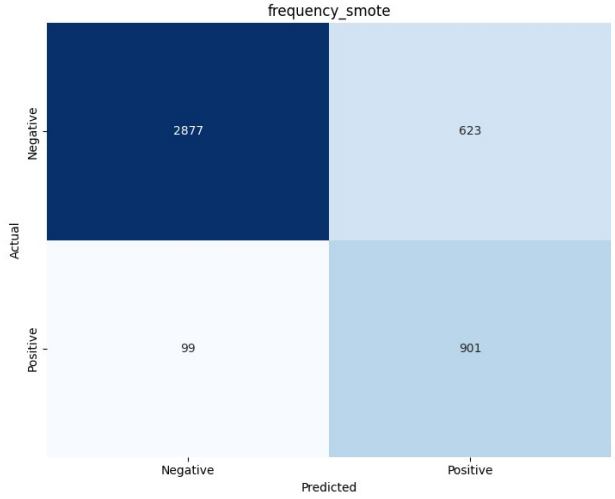
Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63



Metrics

Accuracy: 0.77
Precision: 0.49
Recall (Sensitivity): 0.81
Specificity: 0.76
F1 Score: 0.61

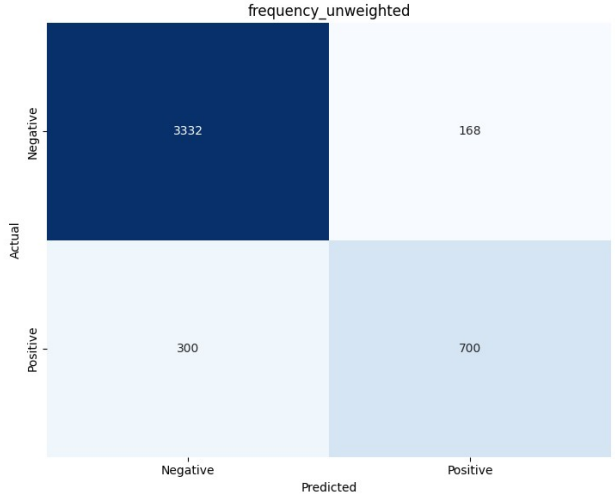
frequency_smote



Metrics

Accuracy: 0.84
Precision: 0.59
Recall (Sensitivity): 0.90
Specificity: 0.82
F1 Score: 0.71

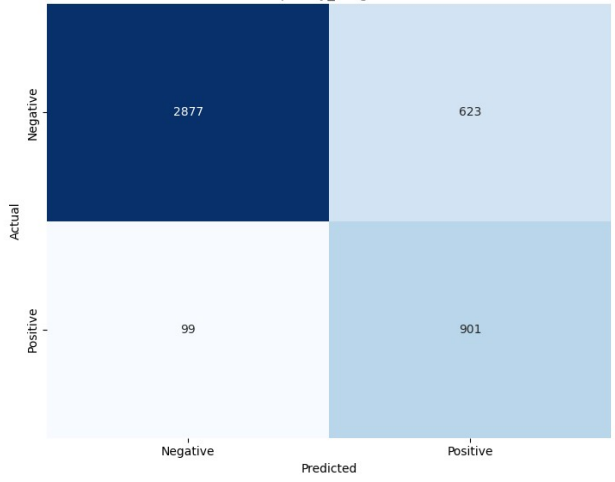
frequency_unweighted



Metrics

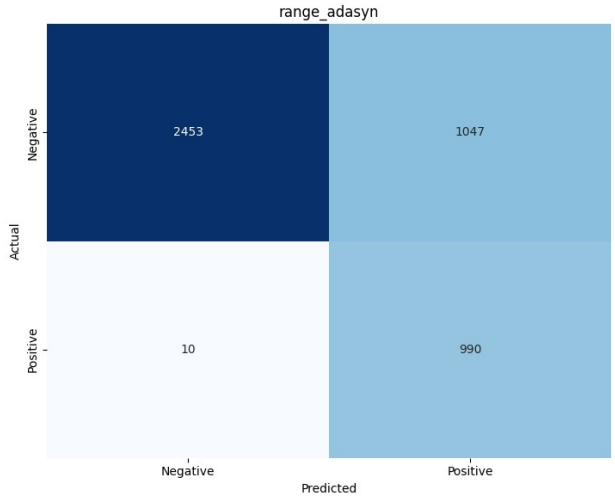
Accuracy: 0.90
Precision: 0.81
Recall (Sensitivity): 0.70
Specificity: 0.95
F1 Score: 0.75

frequency_weighted



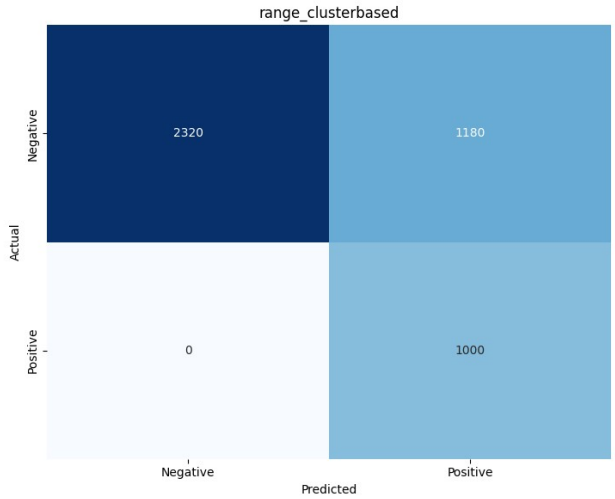
Metrics

Accuracy: 0.84
Precision: 0.59
Recall (Sensitivity): 0.90
Specificity: 0.82
F1 Score: 0.71



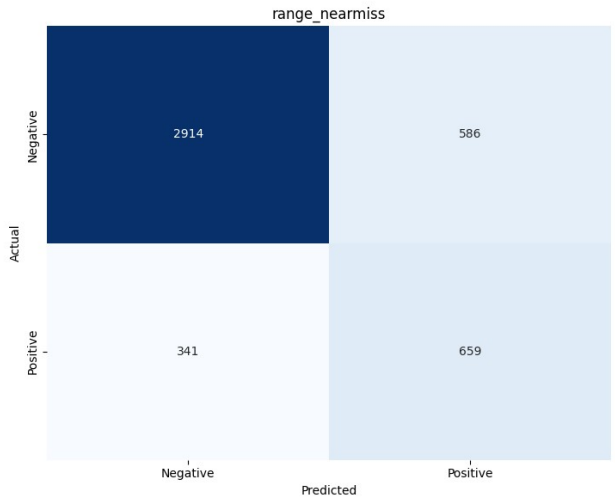
Metrics

Accuracy: 0.77
Precision: 0.49
Recall (Sensitivity): 0.99
Specificity: 0.70
F1 Score: 0.65



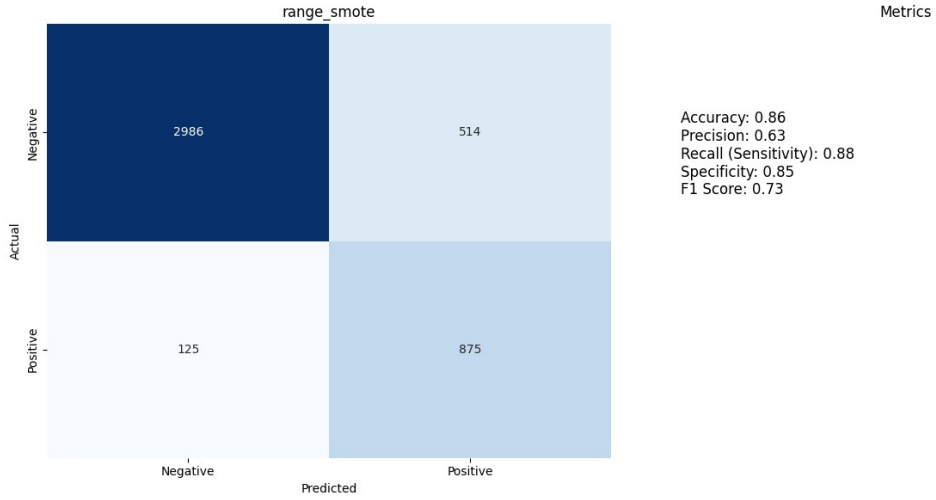
Metrics

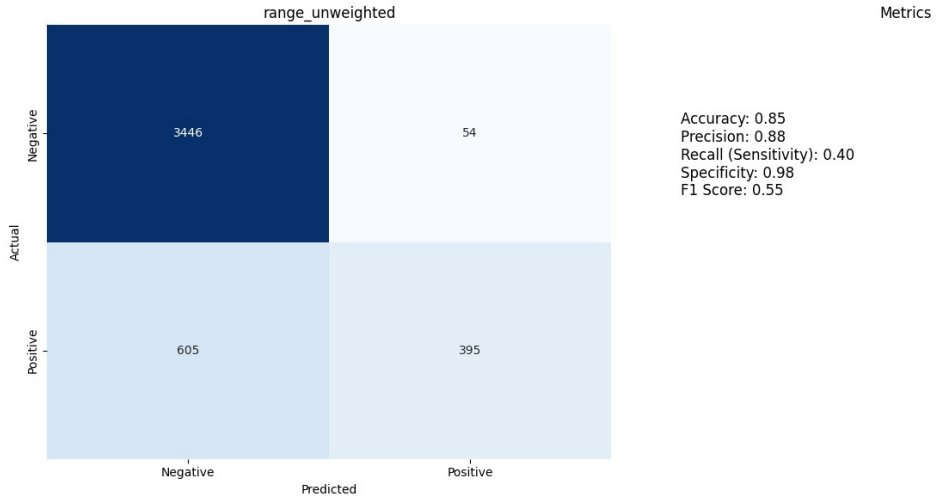
Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63

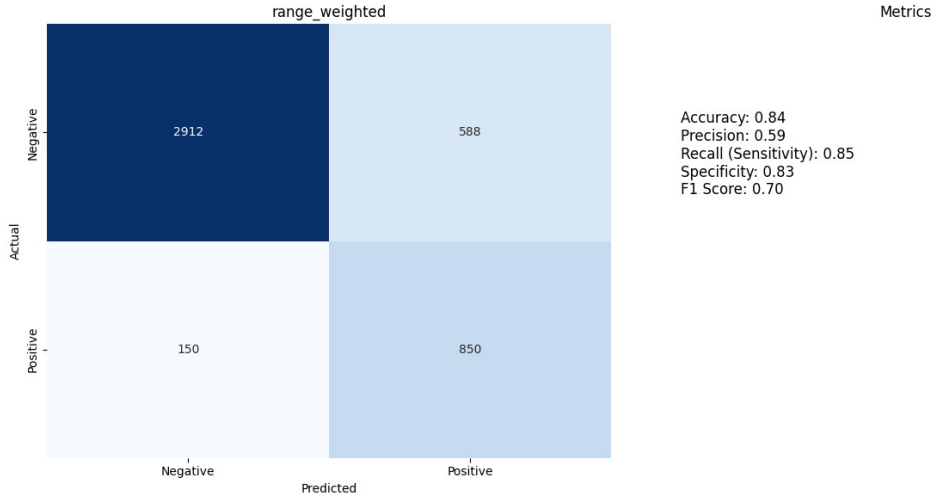


Metrics

Accuracy: 0.79
Precision: 0.53
Recall (Sensitivity): 0.66
Specificity: 0.83
F1 Score: 0.59

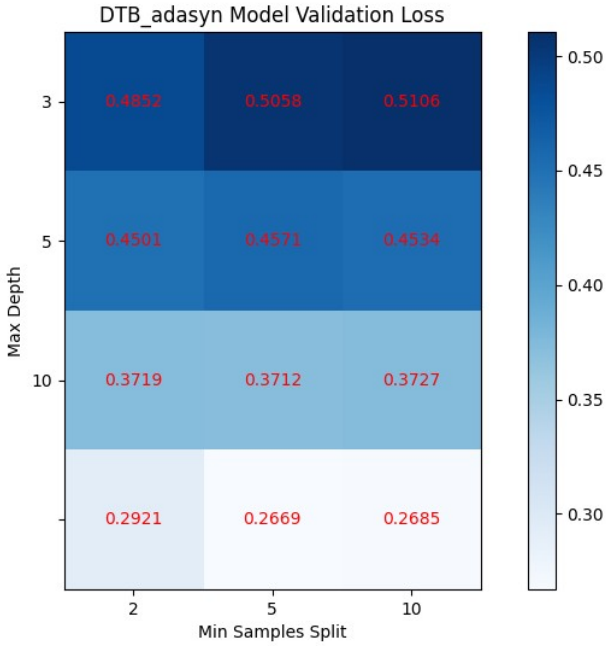


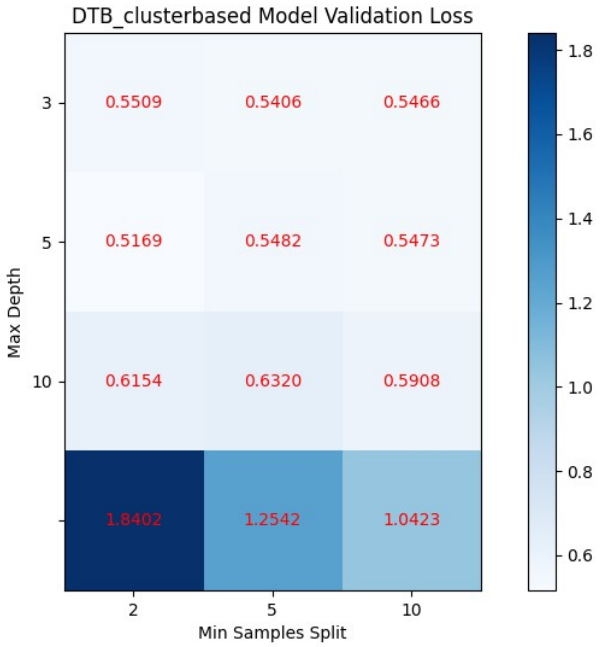


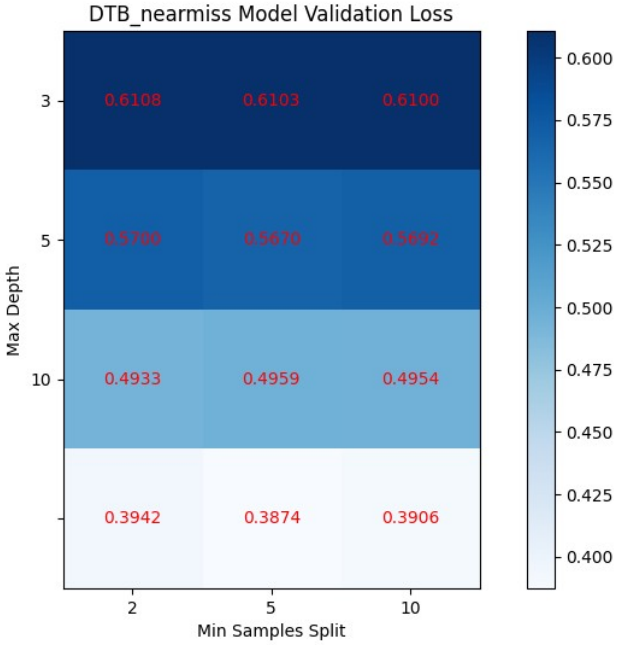


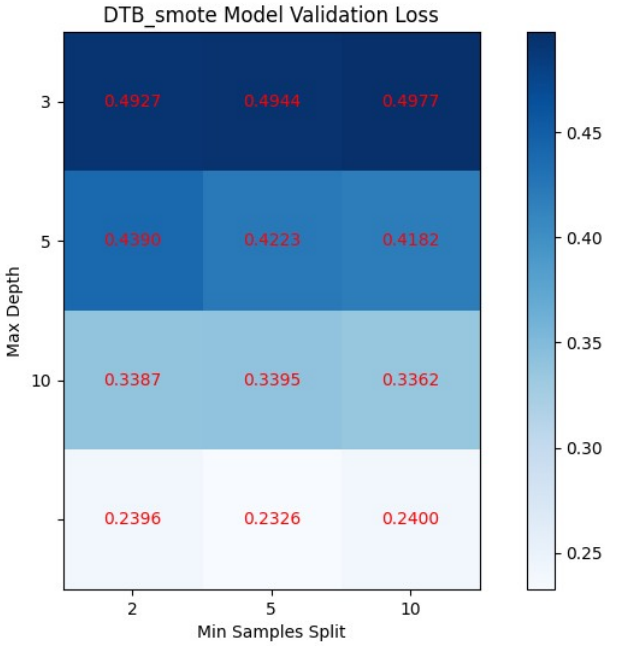
We applied RandomForest which is an ensemble of Decision Trees with bagging. We used 3 different binning. Equal frequency, equal width and decision tree based binning and 5 different datasets Those datasets are created by applying ADASYN, SMOTE, cluster-based undersampling ,NearMiss. The fifth one is the original dataset's proper conversion and we used this set twice, applying weight and unweighted version. We want to compare 18

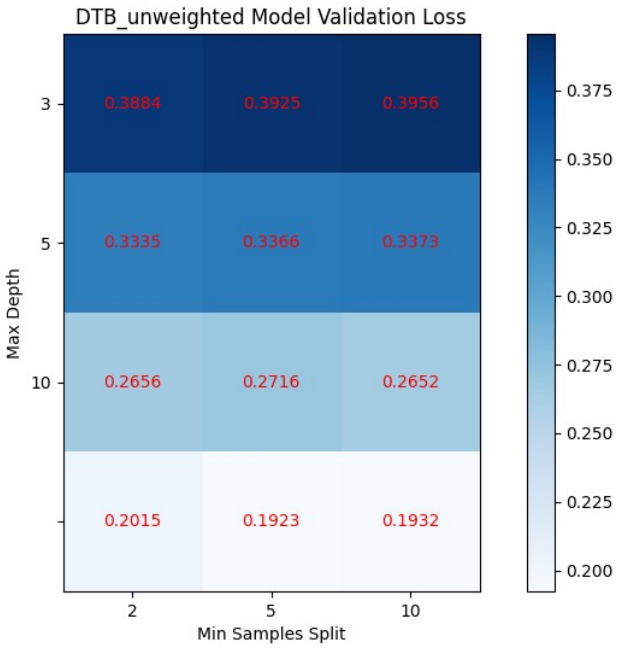
different models for each different dataset. We conducted 216 experiments, 12 for each model to get optimum hyperparameters.Following figures are loss matrices.

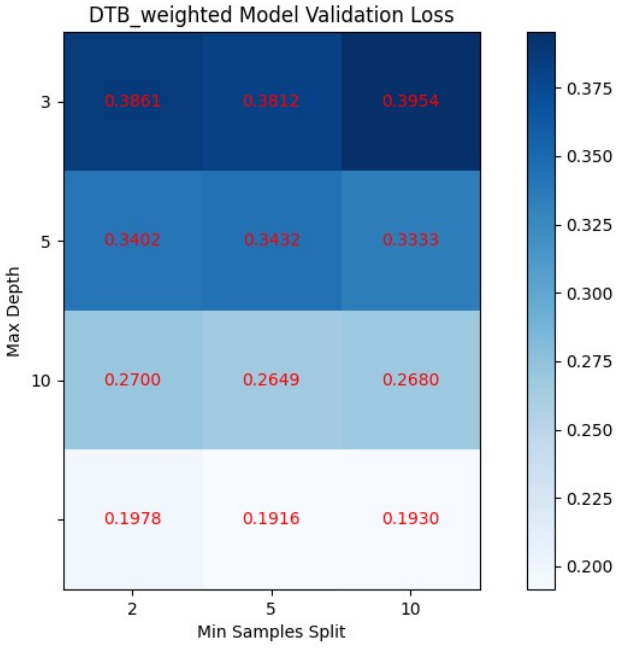


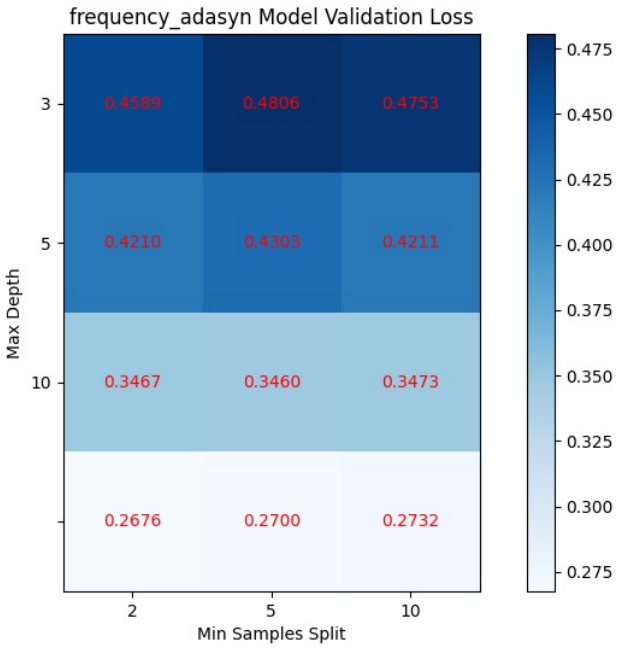


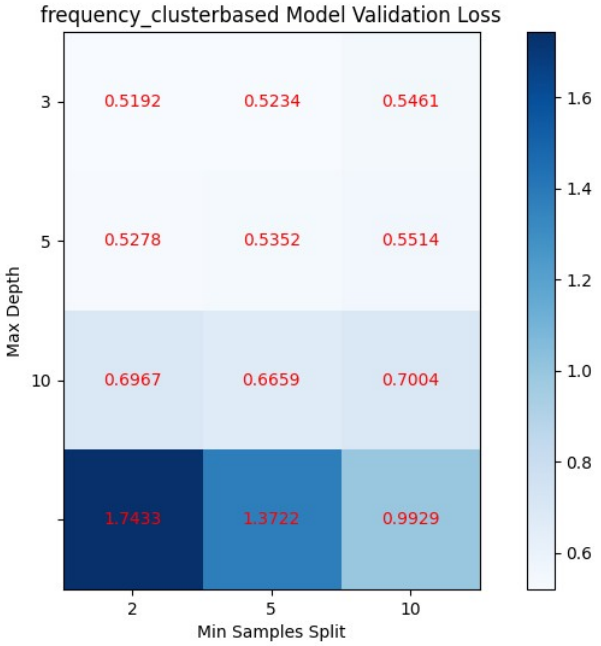

DTB_nearmiss Model Validation Loss
0.6108
0.6103
0.6100
0.5700
0.5670
0.5692
0.4933
0.4959
0.4954
0.3942
0.3874
0.3906
3
5
10
Max Depth
2
5
10
Min Samples Split
0.600
0.575
0.550
0.525
0.500
0.475
0.450
0.425
0.400

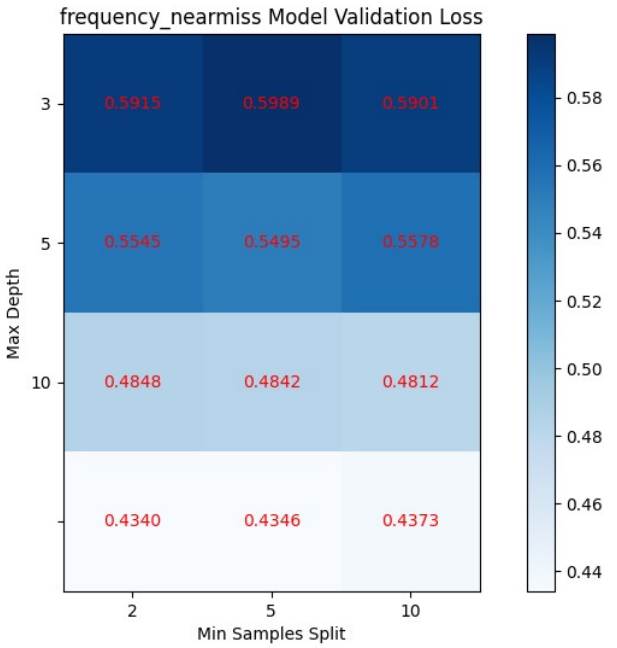

DTB_smote Model Validation Loss
0.4927
0.4944
0.4977
0.4390
0.4223
0.4182
0.3387
0.3395
0.3362
0.2396
0.2326
0.2400
3
5
10
Max Depth
2
5
10
Min Samples Split
0.45
0.40
0.35
0.30
0.25

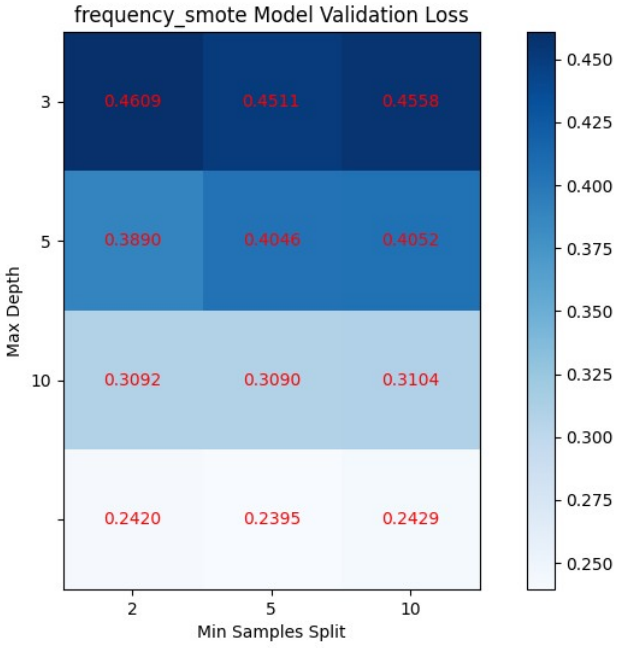

DTB_unweighted Model Validation Loss
0.3884
0.3925
0.3956
0.3335
0.3366
0.3373
0.2656
0.2716
0.2652
0.2015
0.1923
0.1932
3
5
10
Max Depth
2
5
10
Min Samples Split
0.375
0.350
0.325
0.300
0.275
0.250
0.225
0.200

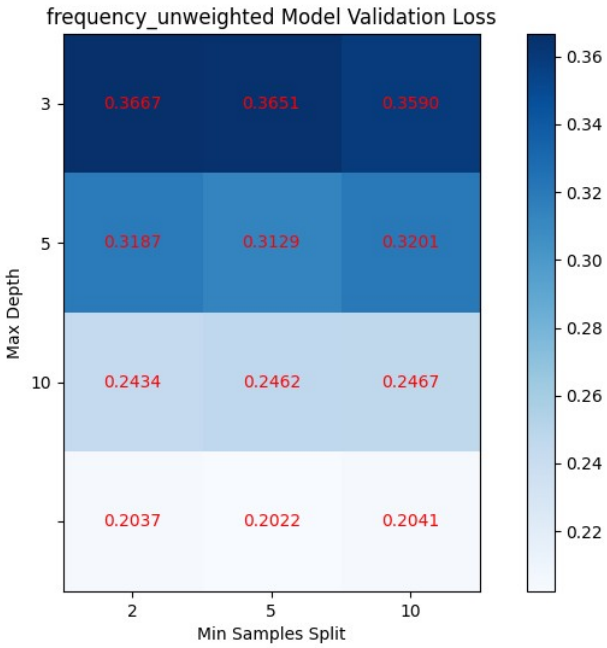

DTB_weighted Model Validation Loss
0.3861
0.3812
0.3954
0.3402
0.3432
0.3333
0.2700
0.2649
0.2680
0.1978
0.1916
0.1930
3
5
10
Max Depth
2
5
10
Min Samples Split
0.375
0.350
0.325
0.300
0.275
0.250
0.225
0.200

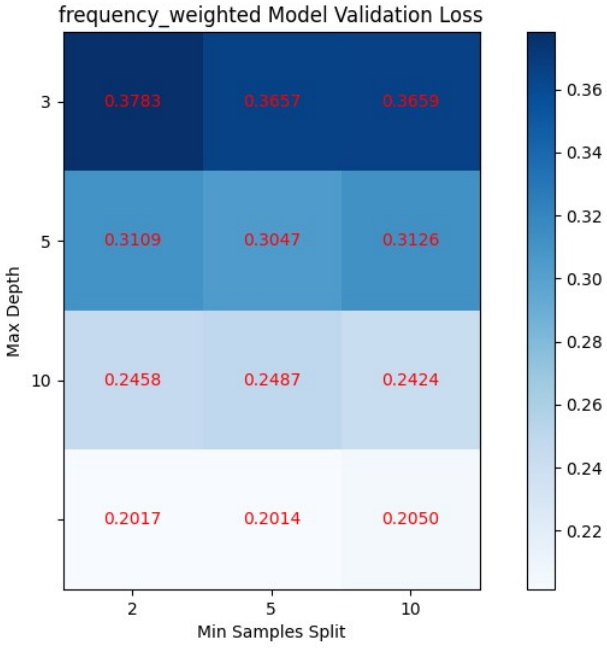

frequency_adasyn Model Validation Loss
0.4589
0.4806
0.4753
0.4210
0.4303
0.4211
0.3467
0.3460
0.3473
0.2676
0.2700
0.2732
Max Depth
3
5
10
2
5
10
Min Samples Split
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0.300
0.275

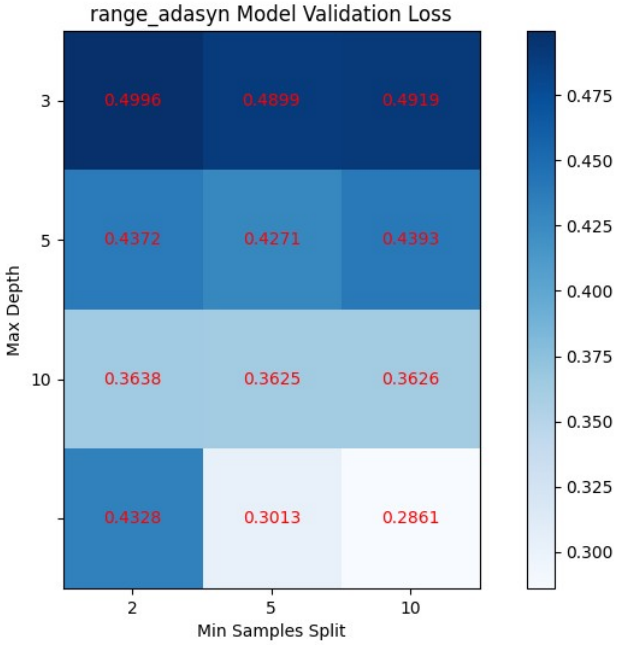

frequency_clusterbased Model Validation Loss
0.5192
0.5234
0.5461
0.5278
0.5352
0.5514
0.6967
0.6659
0.7004
1.7433
1.3722
0.9929
Max Depth
3
5
10
2
5
10
Min Samples Split
1.6
1.4
1.2
1.0
0.8
0.6

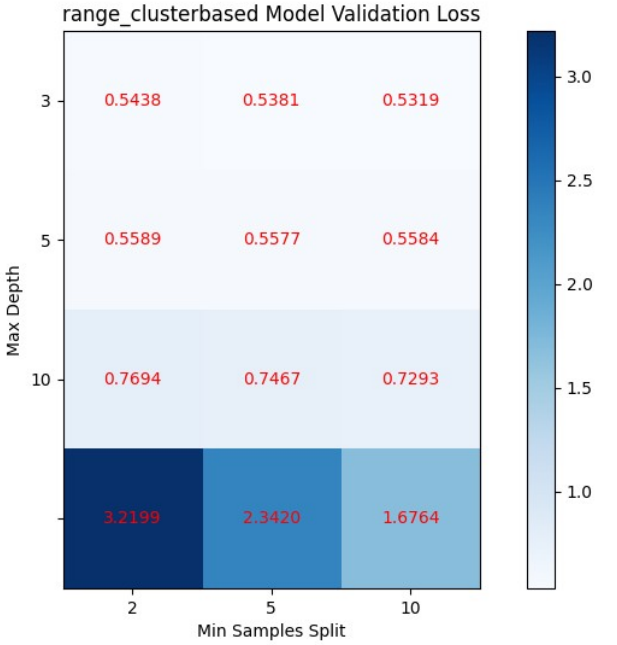

frequency_nearmiss Model Validation Loss
0.5915
0.5989
0.5901
0.5545
0.5495
0.5578
0.4848
0.4842
0.4812
0.4340
0.4346
0.4373
3
5
10
Max Depth
2
5
10
Min Samples Split
0.58
0.56
0.54
0.52
0.50
0.48
0.46
0.44

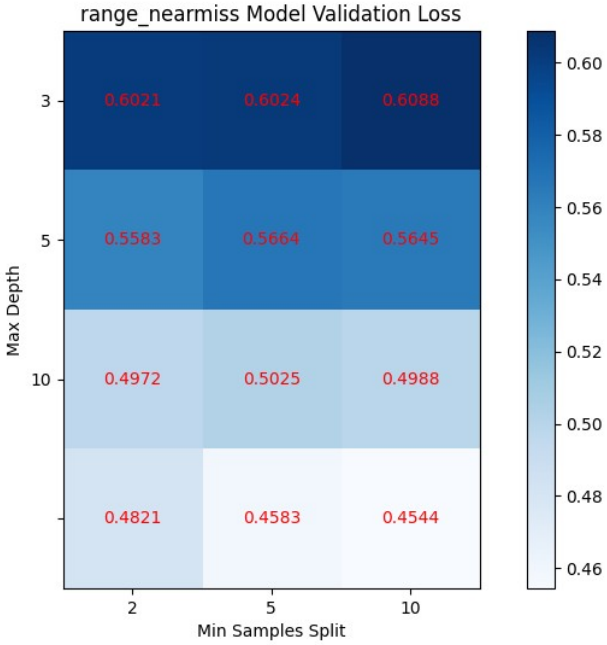

frequency_smote Model Validation Loss
0.4609
0.4511
0.4558
0.3890
0.4046
0.4052
0.3092
0.3090
0.3104
0.2420
0.2395
0.2429
3
5
10
Max Depth
2
5
10
Min Samples Split
0.450
0.425
0.400
0.375
0.350
0.325
0.300
0.275
0.250

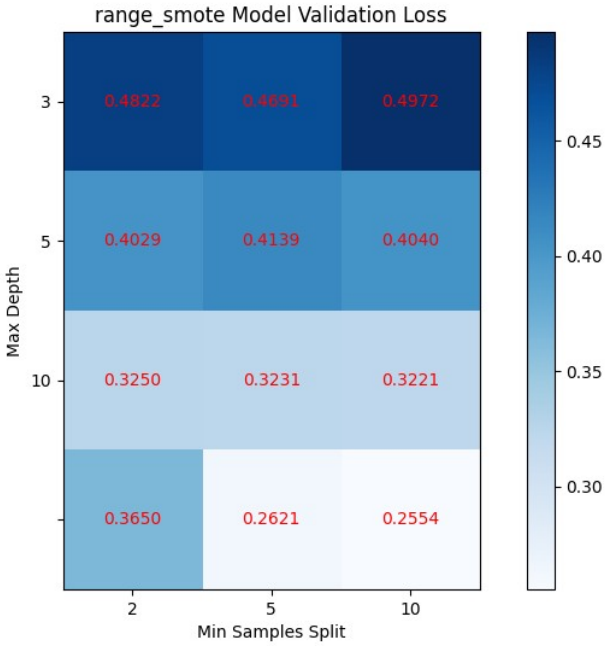

frequency_unweighted Model Validation Loss
0.3667
0.3651
0.3590
0.3187
0.3129
0.3201
0.2434
0.2462
0.2467
0.2037
0.2022
0.2041
3
5
10
Max Depth
2
5
10
Min Samples Split
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22

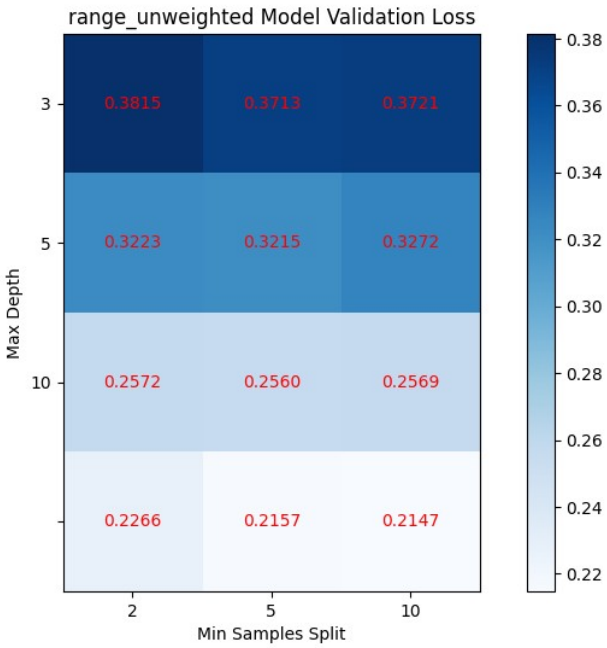

frequency_weighted Model Validation Loss
0.3783
0.3657
0.3659
0.3109
0.3047
0.3126
0.2458
0.2487
0.2424
0.2017
0.2014
0.2050
3
5
10
Max Depth
2
5
10
Min Samples Split
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22

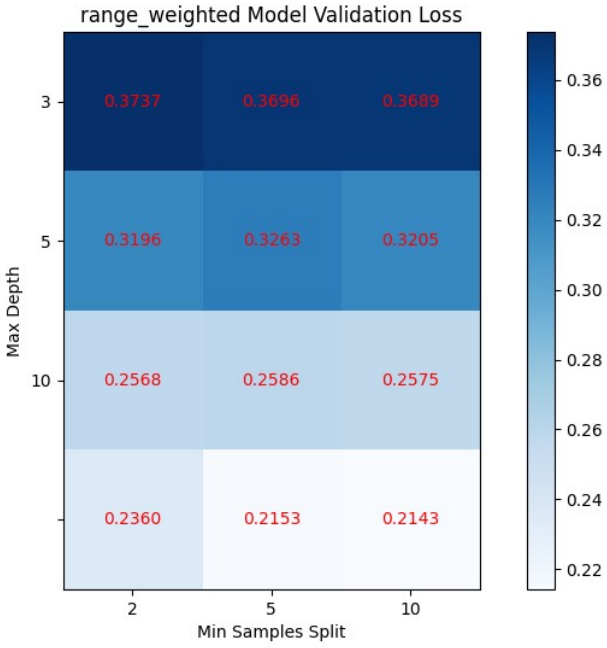

range_adasyn Model Validation Loss
0.4996
0.4899
0.4919
0.4372
0.4271
0.4393
0.3638
0.3625
0.3626
0.4328
0.3013
0.2861
Max Depth
3
5
10
2
5
10
Min Samples Split
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0.300


range_clusterbased Model Validation Loss
0.5438
0.5381
0.5319
0.5589
0.5577
0.5584
0.7694
0.7467
0.7293
3.2199
2.3420
1.6764
Max Depth
3
5
10
2
5
10
Min Samples Split
3.0
2.5
2.0
1.5
1.0

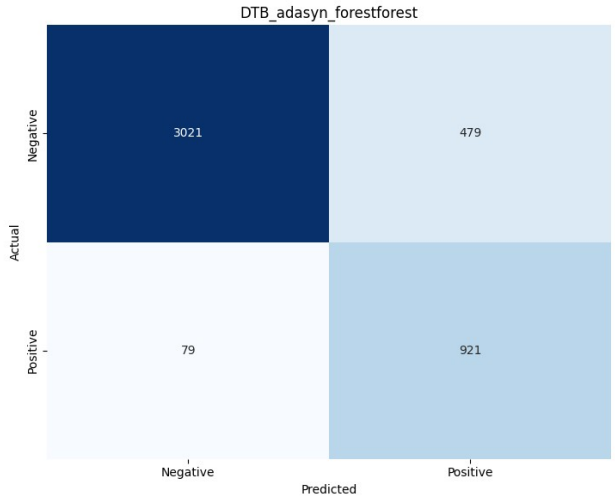

range_nearmiss Model Validation Loss
0.6021
0.6024
0.6088
0.5583
0.5664
0.5645
0.4972
0.5025
0.4988
0.4821
0.4583
0.4544
3
5
10
Max Depth
2
5
10
Min Samples Split
0.60
0.58
0.56
0.54
0.52
0.50
0.48
0.46


range_smote Model Validation Loss
0.4822
0.4691
0.4972
0.4029
0.4139
0.4040
0.3250
0.3231
0.3221
0.3650
0.2621
0.2554
3
5
10
Max Depth
2
5
10
Min Samples Split
0.45
0.40
0.35
0.30


range_unweighted Model Validation Loss
0.3815
0.3713
0.3721
0.3223
0.3215
0.3272
0.2572
0.2560
0.2569
0.2266
0.2157
0.2147
3
5
10
Max Depth
2
5
10
Min Samples Split
0.38
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22

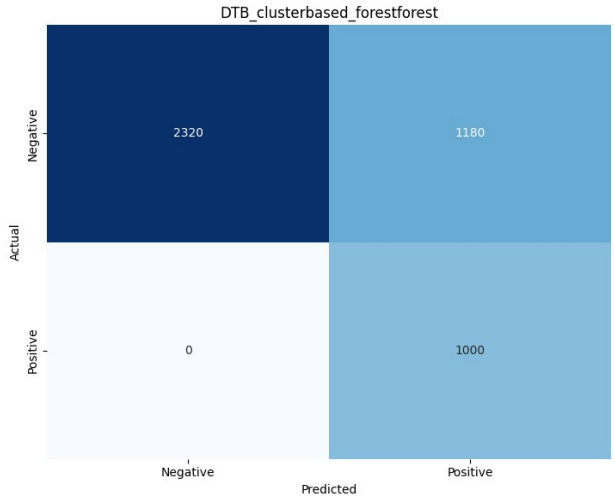

range_weighted Model Validation Loss
0.3737
0.3696
0.3689
0.3196
0.3263
0.3205
0.2568
0.2586
0.2575
0.2360
0.2153
0.2143
3
5
10
Max Depth
2
5
10
Min Samples Split
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22

Confusion matrix and metric results of Random Forest



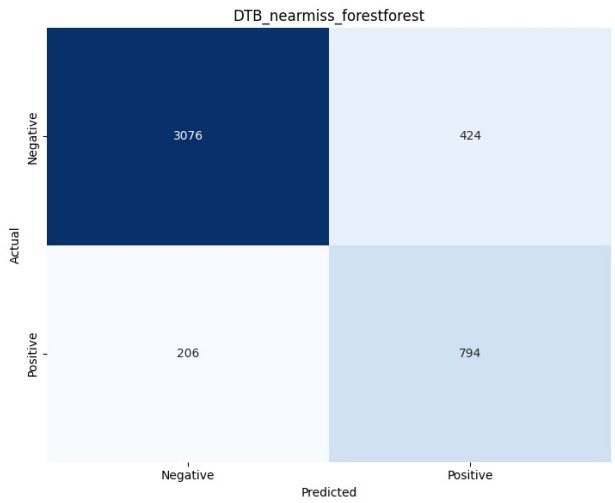
Metrics

Accuracy: 0.88
Precision: 0.66
Recall (Sensitivity): 0.92
Specificity: 0.86
F1 Score: 0.77



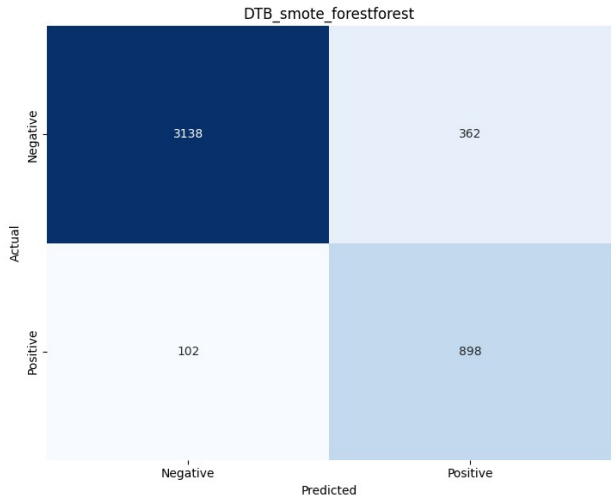
Metrics

Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63



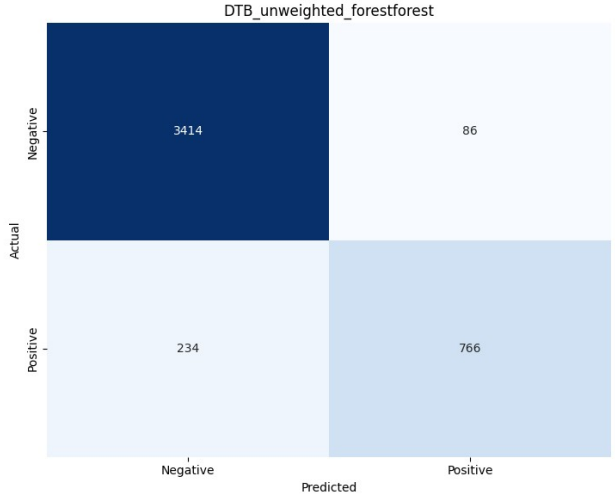
Metrics

Accuracy: 0.86
Precision: 0.65
Recall (Sensitivity): 0.79
Specificity: 0.88
F1 Score: 0.72



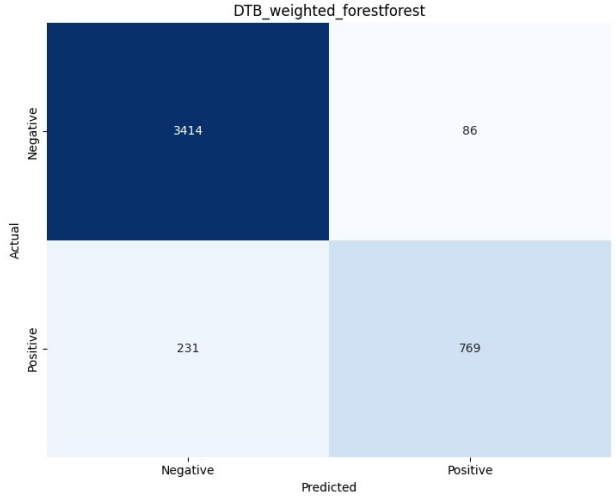
Metrics

Accuracy: 0.90
Precision: 0.71
Recall (Sensitivity): 0.90
Specificity: 0.90
F1 Score: 0.79



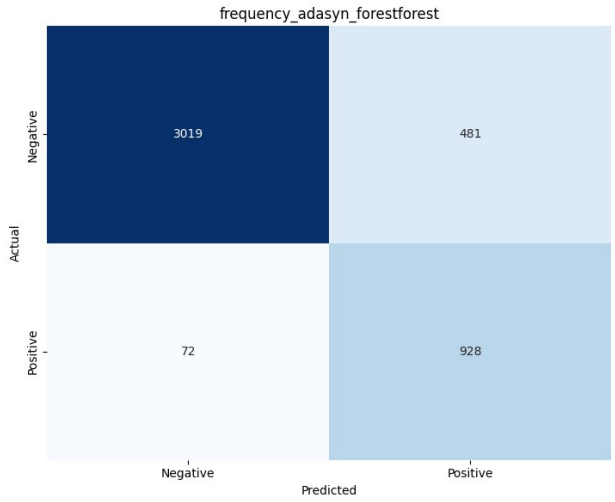
Metrics

Accuracy: 0.93
Precision: 0.90
Recall (Sensitivity): 0.77
Specificity: 0.98
F1 Score: 0.83



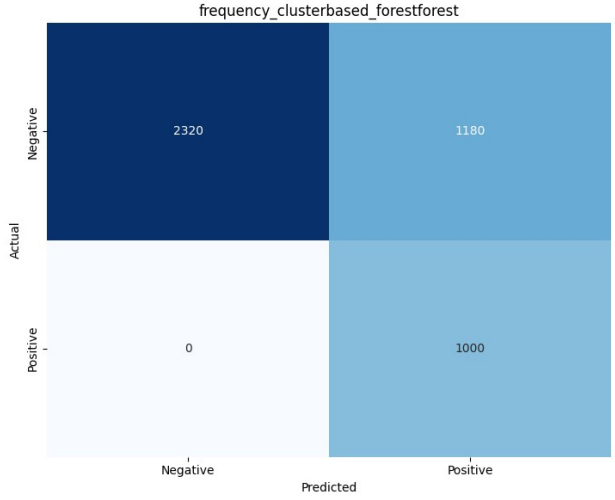
Metrics

Accuracy: 0.93
Precision: 0.90
Recall (Sensitivity): 0.77
Specificity: 0.98
F1 Score: 0.83



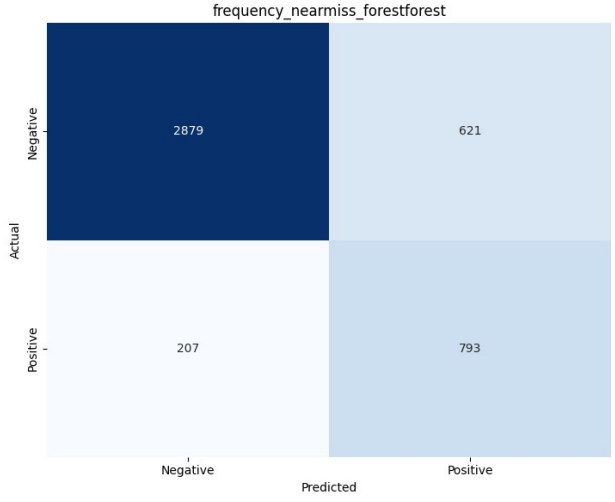
Metrics

Accuracy: 0.88
Precision: 0.66
Recall (Sensitivity): 0.93
Specificity: 0.86
F1 Score: 0.77



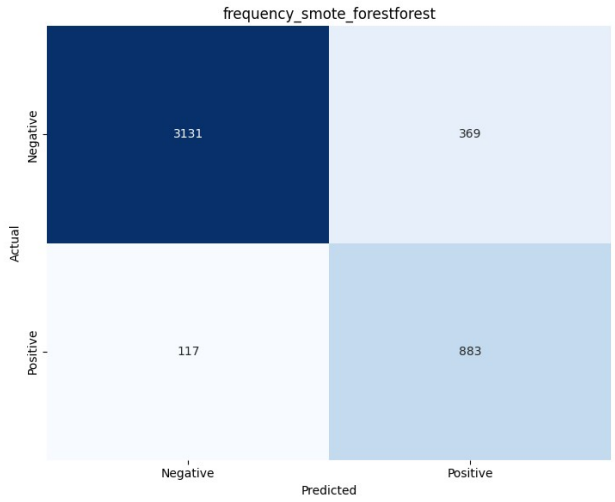
Metrics

Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63



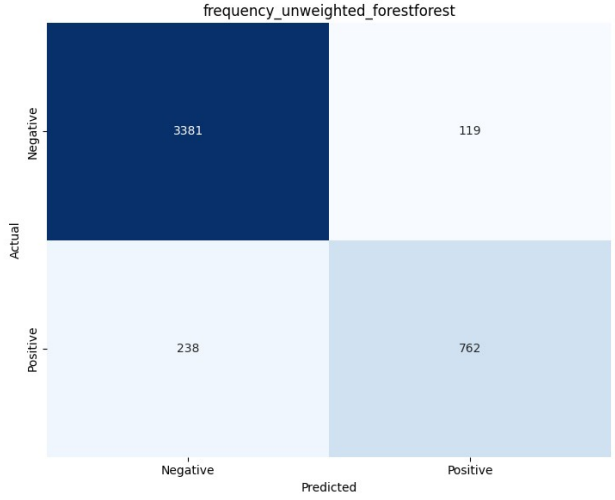
Metrics

Accuracy: 0.82
Precision: 0.56
Recall (Sensitivity): 0.79
Specificity: 0.82
F1 Score: 0.66



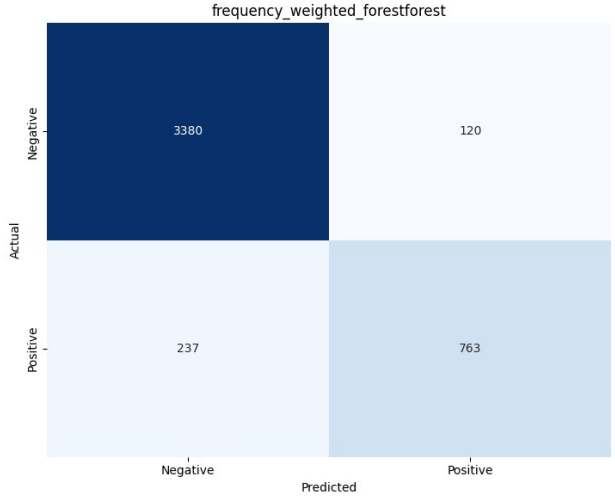
Metrics

Accuracy: 0.89
Precision: 0.71
Recall (Sensitivity): 0.88
Specificity: 0.89
F1 Score: 0.78



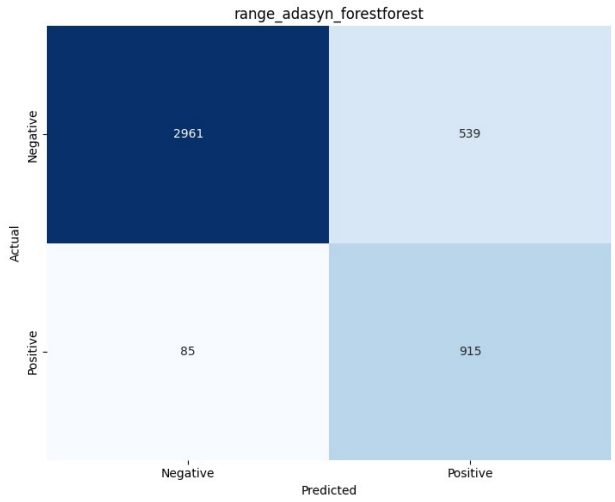
Metrics

Accuracy: 0.92
Precision: 0.86
Recall (Sensitivity): 0.76
Specificity: 0.97
F1 Score: 0.81



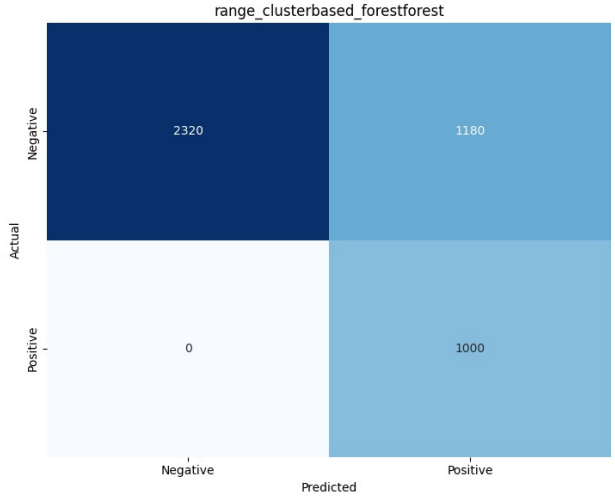
Metrics

Accuracy: 0.92
Precision: 0.86
Recall (Sensitivity): 0.76
Specificity: 0.97
F1 Score: 0.81



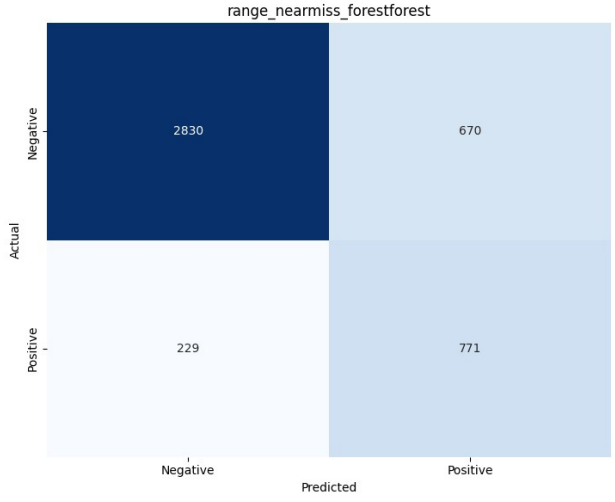
Metrics

Accuracy: 0.86
Precision: 0.63
Recall (Sensitivity): 0.92
Specificity: 0.85
F1 Score: 0.75



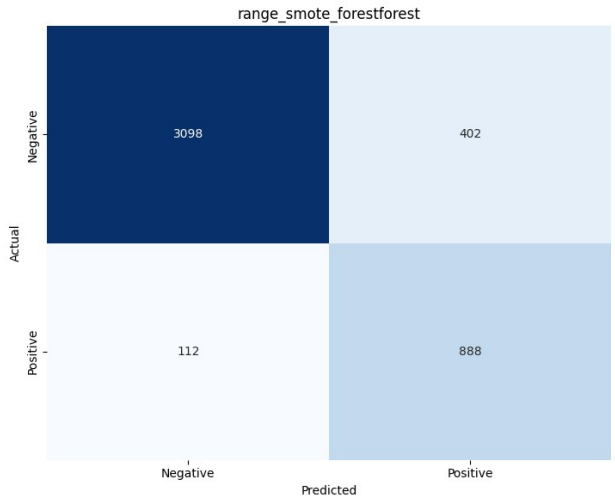
Metrics

Accuracy: 0.74
Precision: 0.46
Recall (Sensitivity): 1.00
Specificity: 0.66
F1 Score: 0.63



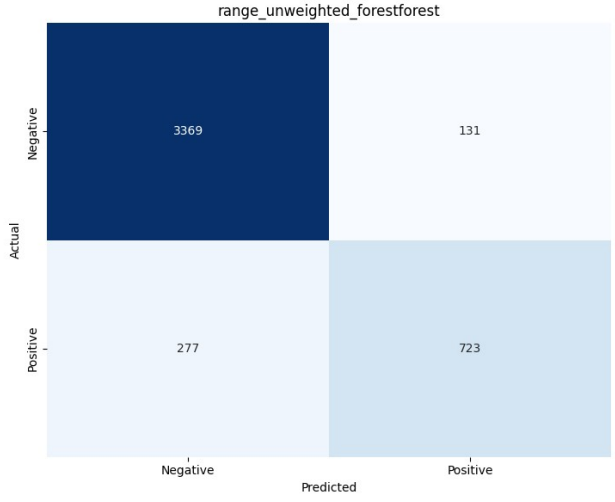
Metrics

Accuracy: 0.80
Precision: 0.54
Recall (Sensitivity): 0.77
Specificity: 0.81
F1 Score: 0.63



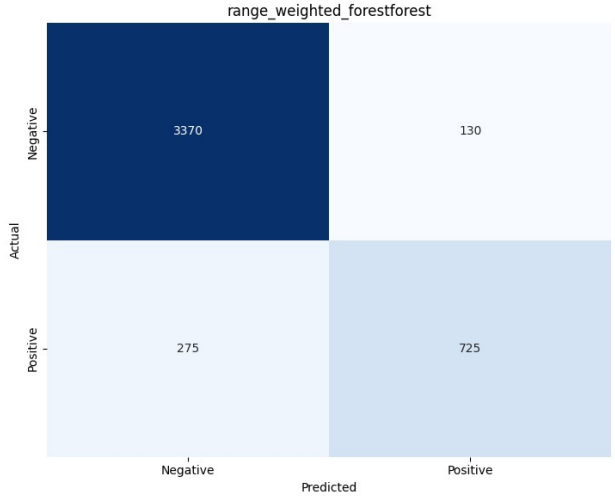
Metrics

Accuracy: 0.89
Precision: 0.69
Recall (Sensitivity): 0.89
Specificity: 0.89
F1 Score: 0.78



Metrics

Accuracy: 0.91
Precision: 0.85
Recall (Sensitivity): 0.72
Specificity: 0.96
F1 Score: 0.78



Metrics

Accuracy: 0.91
Precision: 0.85
Recall (Sensitivity): 0.72
Specificity: 0.96
F1 Score: 0.78

Voting Classifier

The primary goal of the Voting Evaluator is to enhance predictive accuracy and mitigate the weaknesses of individual classifiers. It achieves this by employing an ensemble method known as a voting classifier, which consolidates predictions from diverse models. The system operates using a "soft voting" approach, where each model contributes probabilistic predictions that are aggregated based on predefined weights. Unlike soft voting, hard voting does not rely on weighted probabilistic predictions. This approach uses majority voting. In a situation such as tie break, the ultimate decision is decided by the first model implemented in VotingClassifier. This strategy allows the ensemble to make more informed decisions, particularly in scenarios involving imbalanced or complex datasets.

Model Integration

The evaluator incorporates a diverse set of machine learning models, including:

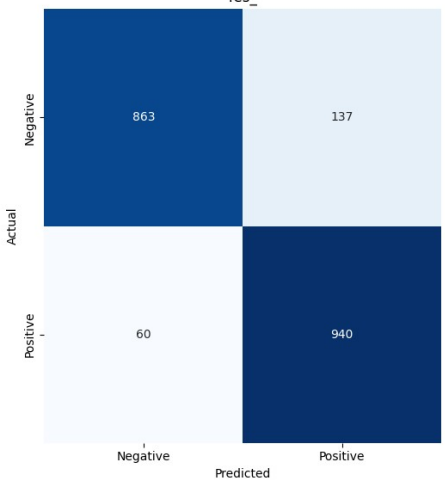
- **Artificial Neural Networks (ANNs)**: Used for their capacity to model non-linear relationships and handle high-dimensional data.
- **Decision Trees and Random Forests**: Known for their interpretability and ability to capture feature interactions.
- **Naive Bayes Models**: Valued for their simplicity and effectiveness in probabilistic predictions, particularly in text or categorical data.
- **Boosted Naive Bayes**: An enhanced version of Naive Bayes that uses boosting techniques to improve model performance on difficult samples.

Advantages

1. **Improved Accuracy**: By combining multiple models, the ensemble typically outperforms individual classifiers in terms of predictive accuracy.
2. **Robustness**: The diversity of integrated models ensures resilience against weaknesses of any single algorithm.
3. **Interpretability**: The use of weights in the soft voting mechanism allows for controlled influence from each model, offering insights into the contribution of each algorithm.
4. **Flexibility**: The system can incorporate new models or adjust weights to adapt to different datasets and requirements.

Results with the best models

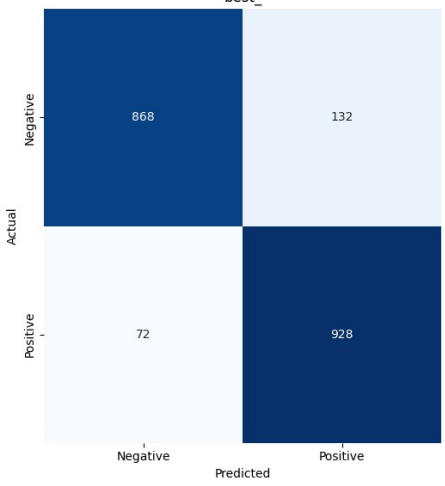
res_



Metrics

Accuracy: 0.90
Precision: 0.87
Recall: 0.94
Specificity: 0.86
F1 Score: 0.91
Models: ['ANN_2', 'rf DTB_smote', 'nb DTB_adasyn', 'ada_nb DTB_smote', 'dt range smote', 'ANN_1
Voting: hard
Weights: None

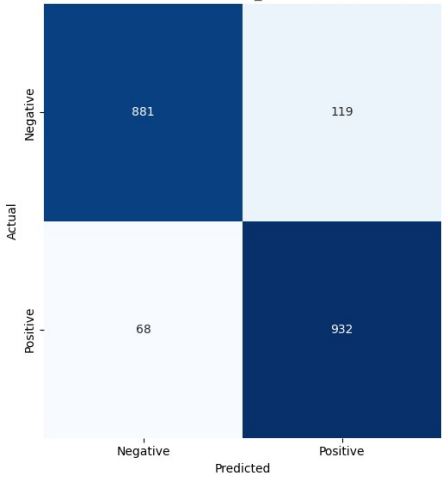
best_



Metrics

Accuracy: 0.90
Precision: 0.88
Recall: 0.93
Specificity: 0.87
F1 Score: 0.90
Models: ['ANN_2', 'rf DTB_smote', 'nb DTB_adasyn', 'ada_nb DTB_smote', 'dt range smote', 'ANN_1
Voting: soft
Weights: None

best_



Metrics

Accuracy: 0.91
Precision: 0.89
Recall: 0.93
Specificity: 0.88
F1 Score: 0.91
Models: ['ANN_2', 'rf DTB_smote', 'nb DTB_adasyn', 'ada_nb DTB_smote', 'dt range smote', 'ANN_1
Voting: soft
Weights: [3, 3, 0.3, 1, 0.05, 0.15]

Discussion

In this study, several machine learning algorithms were applied to address the classification problem, yielding varying outcomes. The initial test showed that precision was lower due to the number of data instances, but as the gap between classes narrowed with undersampling of test set, the F1 score improved. However, the accuracy decreased when the biased model, trained on the original dataset, was tested, as the test data ratio shifted. This drop in accuracy was mitigated in models where class imbalance was addressed through oversampling, undersampling, or weight assignment, with some even showing an increase.Ensemble methods showed more promising results, enhancing both accuracy and robustness. These findings underscore the importance of addressing class imbalance and suggest that ensemble methods are more effective than traditional strategies for this classification problem.