Name: Nahush Jayesh Patil

CS 6375.003 Fall 24 - MACHINE LEARNING PROJECT 1 - SPAM & HAM CLASSIFIER

Experiments

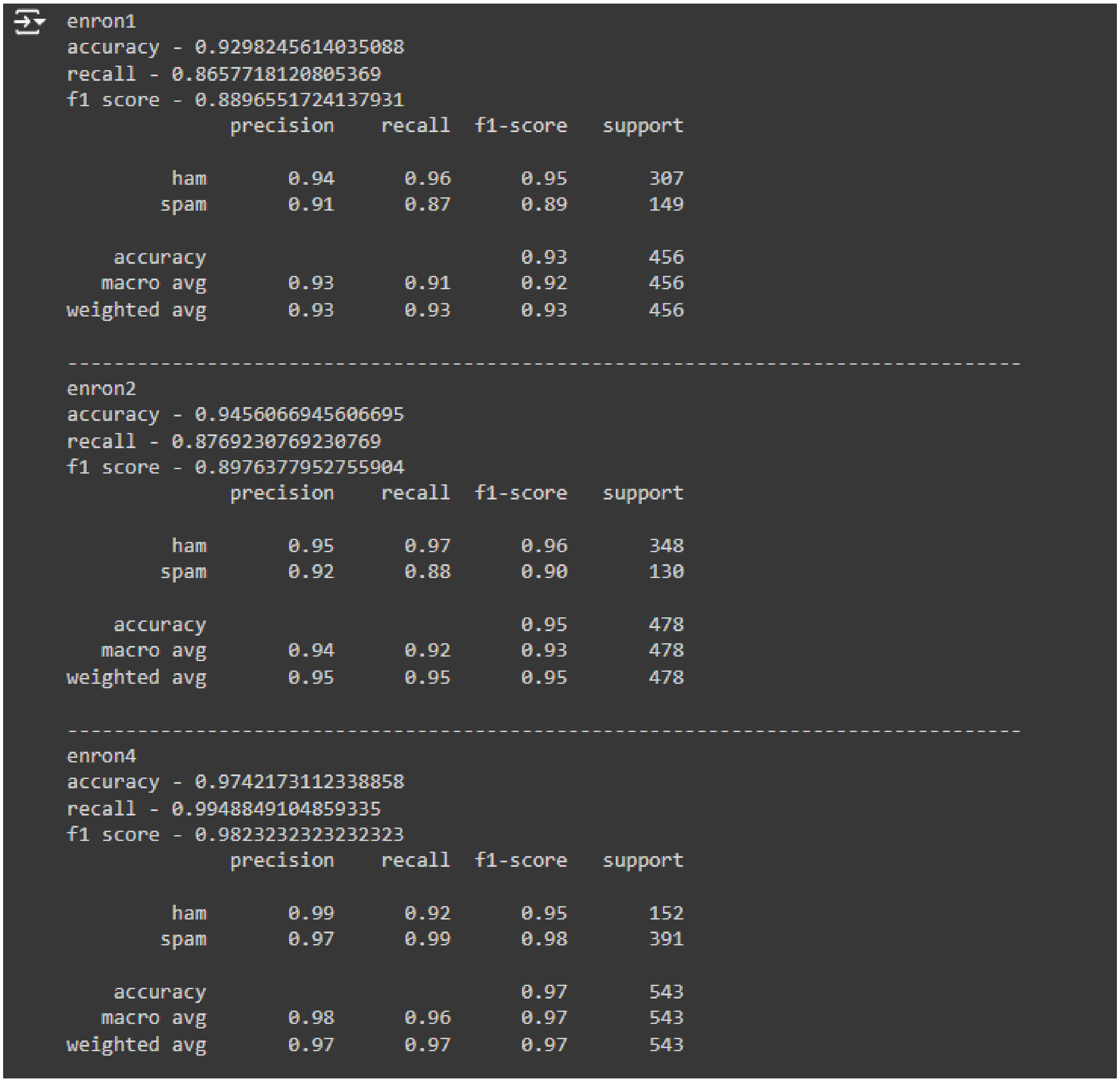
1. Multinomial Naive Bayes for Bag of Words dataset
2. Discrete Naive Bayes for Bernoulli dataset
3. Logistic Regression for Bag of Words dataset
4. Logistic Regression for Bernoulli dataset
5. Stochastic Gradient Descent for Bag of Words dataset
6. Stochastic Gradient Descent for Bernoulli dataset

MULTINOMIAL NAIVE BAYES

Scikit learn approach results -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	92.98%	86.57%	88.96%	91.48%
Enron 2	94.56%	87.69%	89.76%	91.93%
Enron 4	97.42%	99.48%	98.23%	97%

Name: Nahush Jayesh Patil

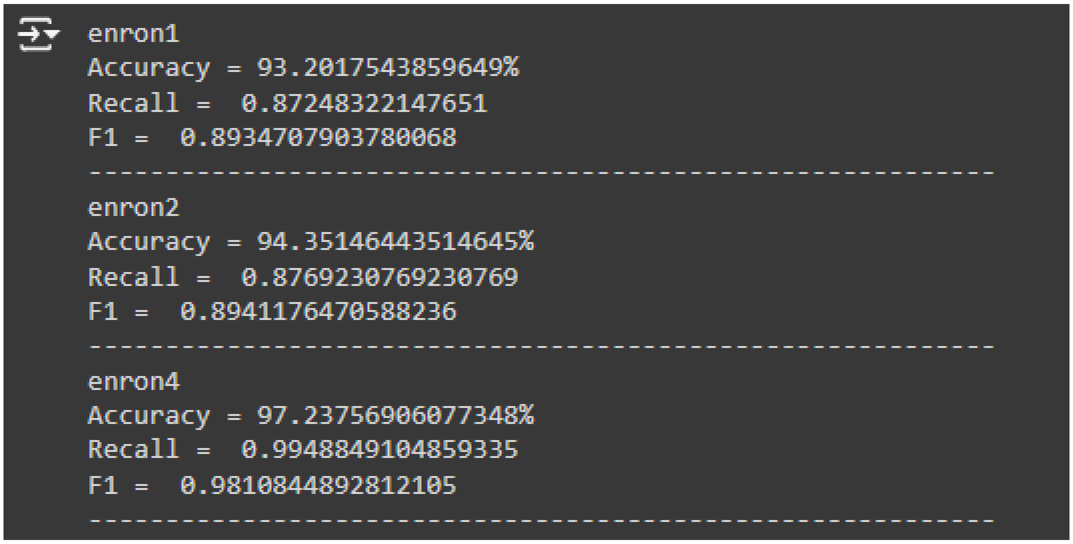


```
enron1
accuracy - 0.9298245614035088
recall - 0.8657718120805369
f1 score - 0.8896551724137931
              precision    recall  f1-score   support

         ham       0.94      0.96      0.95       307
        spam       0.91      0.87      0.89       149

    accuracy                           0.93       456
   macro avg       0.93      0.91      0.92       456
weighted avg       0.93      0.93      0.93       456

----------------------------------------------------------------------------------
enron2
accuracy - 0.9456066945606695
recall - 0.8769230769230769
f1 score - 0.8976377952755904
              precision    recall  f1-score   support

         ham       0.95      0.97      0.96       348
        spam       0.92      0.88      0.90       130

    accuracy                           0.95       478
   macro avg       0.94      0.92      0.93       478
weighted avg       0.95      0.95      0.95       478

----------------------------------------------------------------------------------
enron4
accuracy - 0.9742173112338858
recall - 0.9948849104859335
f1 score - 0.9823232323232323
              precision    recall  f1-score   support

         ham       0.99      0.92      0.95       152
        spam       0.97      0.99      0.98       391

    accuracy                           0.97       543
   macro avg       0.98      0.96      0.97       543
weighted avg       0.97      0.97      0.97       543
```

Step by step approach results -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	93.20%	87.24%	89.34%	91.54%
Enron 2	94.35%	87.69%	89.41%	91.2%
Enron 4	97.23%	99.48%	98.10%	96.76%

Name: Nahush Jayesh Patil



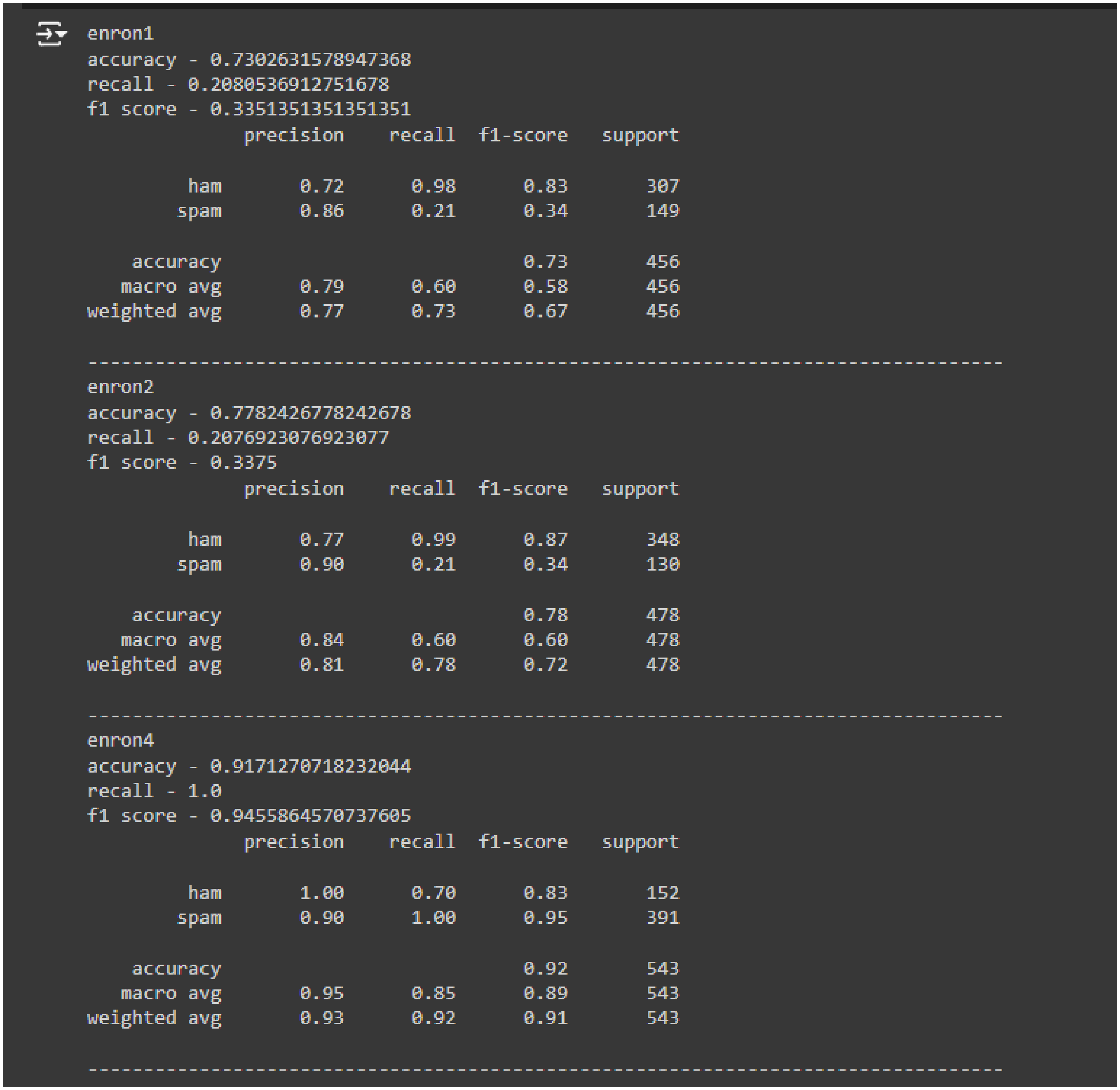
```
enron1
Accuracy = 93.2017543859649%
Recall =  0.87248322147651
F1 =  0.8934707903780068
-----------------------------------------------------------
enron2
Accuracy = 94.35146443514645%
Recall =  0.8769230769230769
F1 =  0.8941176470588236
-----------------------------------------------------------
enron4
Accuracy = 97.23756906077348%
Recall =  0.9948849104859335
F1 =  0.9810844892812105
-----------------------------------------------------------
```

BERNOULLI NAIVE BAYES

Scikit learn approach results -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	73.02%	20.80%	33.51%	97%
Enron 2	77.82%	20.76%	33.75%	97%
Enron 4	91.71%	100%	94.55%	97%

Name: Nahush Jayesh Patil

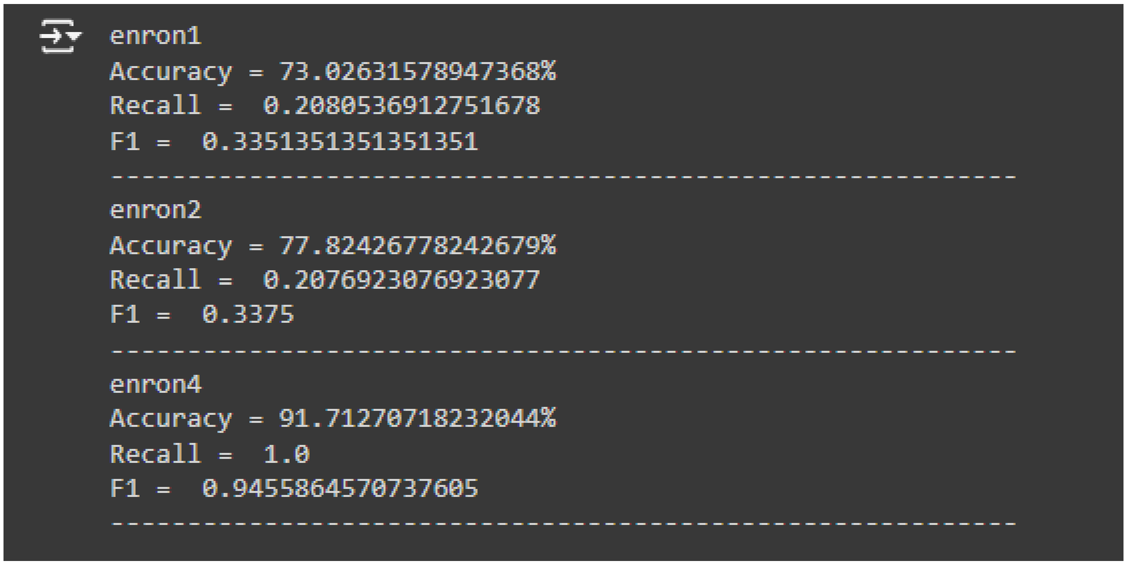


```
enron1
accuracy - 0.7302631578947368
recall - 0.2080536912751678
f1 score - 0.3351351351351351
              precision    recall  f1-score   support

         ham       0.72      0.98      0.83       307
        spam       0.86      0.21      0.34       149

    accuracy                           0.73       456
   macro avg       0.79      0.60      0.58       456
weighted avg       0.77      0.73      0.67       456

----------------------------------------------------------------------------------
enron2
accuracy - 0.7782426778242678
recall - 0.2076923076923077
f1 score - 0.3375
              precision    recall  f1-score   support

         ham       0.77      0.99      0.87       348
        spam       0.90      0.21      0.34       130

    accuracy                           0.78       478
   macro avg       0.84      0.60      0.60       478
weighted avg       0.81      0.78      0.72       478

----------------------------------------------------------------------------------
enron4
accuracy - 0.9171270718232044
recall - 1.0
f1 score - 0.9455864570737605
              precision    recall  f1-score   support

         ham       1.00      0.70      0.83       152
        spam       0.90      1.00      0.95       391

    accuracy                           0.92       543
   macro avg       0.95      0.85      0.89       543
weighted avg       0.93      0.92      0.91       543

----------------------------------------------------------------------------------
```

Name: Nahush Jayesh Patil

Step by step approach results -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	73.02%	20.80%	33.51%	86.11%
Enron 2	77.82%	20.76%	33.75%	90%
Enron 4	91.71%	100%	94.55%	89.67%



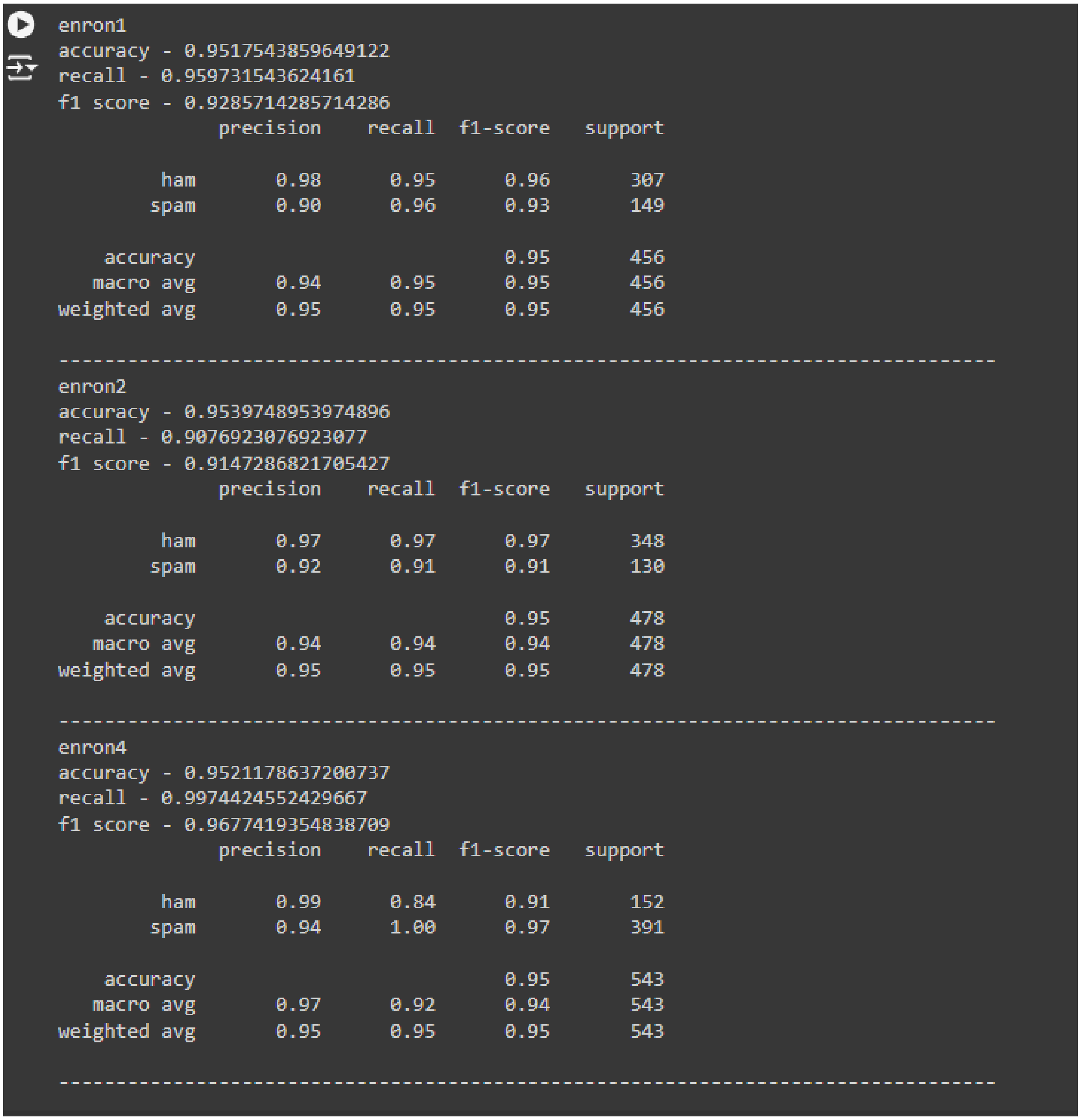
```
enron1
Accuracy = 73.02631578947368%
Recall =  0.2080536912751678
F1 =  0.3351351351351351
-----------------------------------------------------------
enron2
Accuracy = 77.82426778242679%
Recall =  0.2076923076923077
F1 =  0.3375
-----------------------------------------------------------
enron4
Accuracy = 91.71270718232044%
Recall =  1.0
F1 =  0.9455864570737605
-----------------------------------------------------------
```

LOGISTIC REGRESSION -

Scikit learn approach Bag of Words results -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	95.17%	95.97%	92.85%	97%
Enron 2	95.39%	90.76%	91.47%	97%
Enron 4	95.21%	99.74%	96.77%	97%

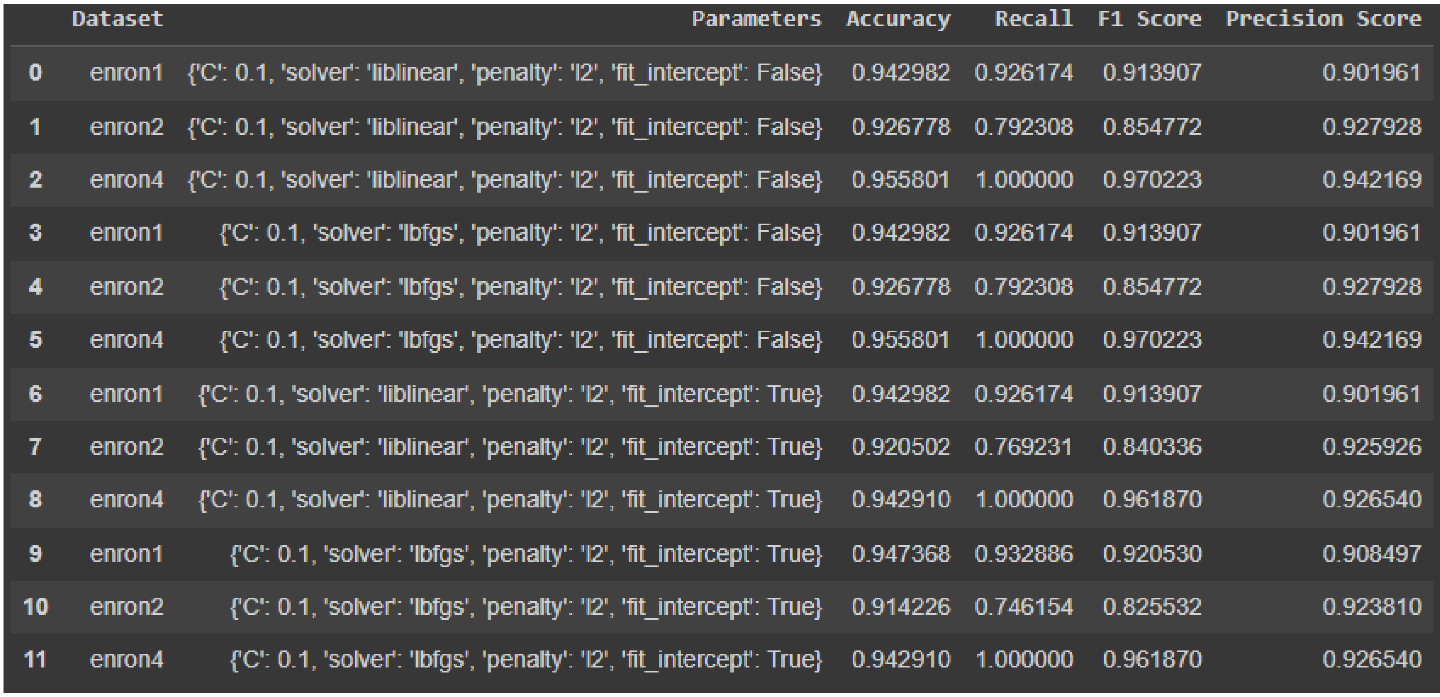
Name: Nahush Jayesh Patil



```
enron1
accuracy - 0.9517543859649122
recall - 0.959731543624161
f1 score - 0.9285714285714286
              precision    recall  f1-score   support

         ham       0.98      0.95      0.96       307
        spam       0.90      0.96      0.93       149

    accuracy                           0.95       456
   macro avg       0.94      0.95      0.95       456
weighted avg       0.95      0.95      0.95       456

----------------------------------------------------------------------------------
enron2
accuracy - 0.9539748953974896
recall - 0.9076923076923077
f1 score - 0.9147286821705427
              precision    recall  f1-score   support

         ham       0.97      0.97      0.97       348
        spam       0.92      0.91      0.91       130

    accuracy                           0.95       478
   macro avg       0.94      0.94      0.94       478
weighted avg       0.95      0.95      0.95       478

----------------------------------------------------------------------------------
enron4
accuracy - 0.9521178637200737
recall - 0.9974424552429667
f1 score - 0.9677419354838709
              precision    recall  f1-score   support

         ham       0.99      0.84      0.91       152
        spam       0.94      1.00      0.97       391

    accuracy                           0.95       543
   macro avg       0.97      0.92      0.94       543
weighted avg       0.95      0.95      0.95       543

----------------------------------------------------------------------------------
```

Name: Nahush Jayesh Patil

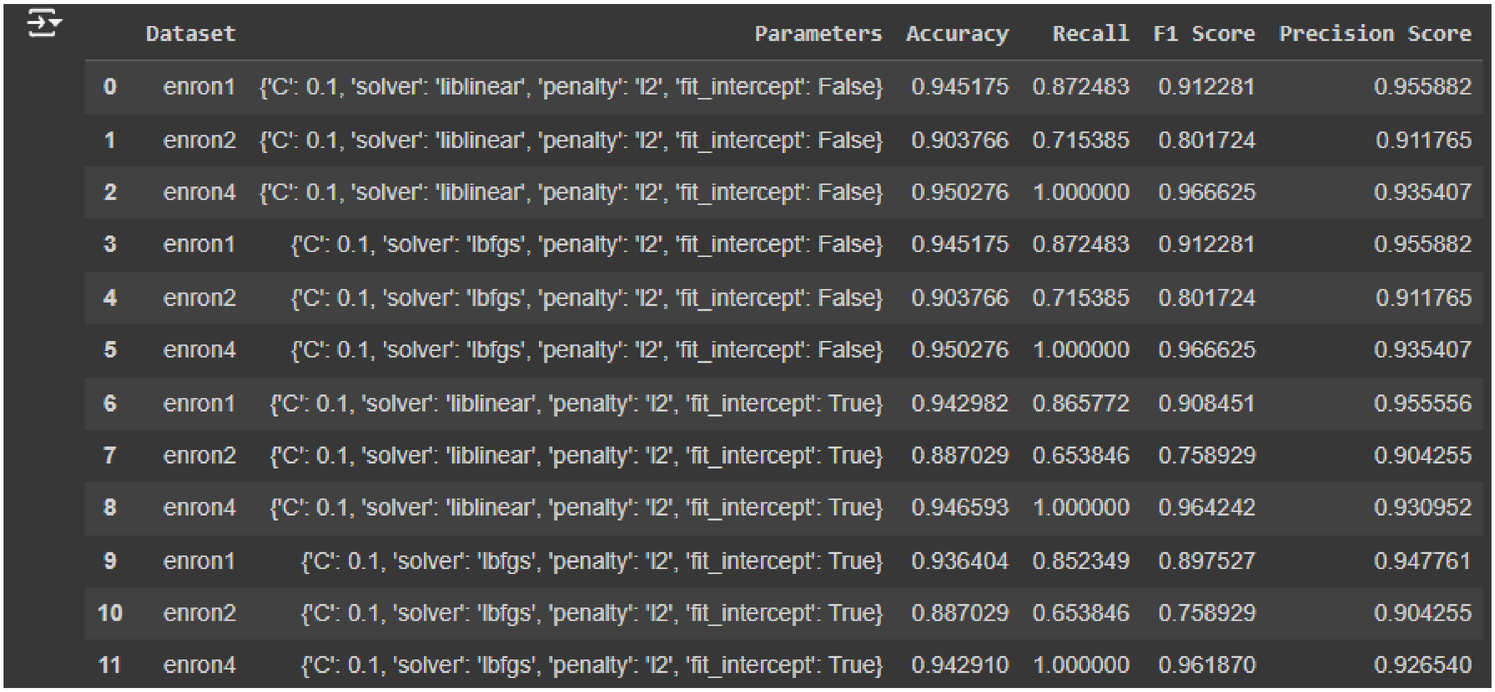
Hyperparameter tuned BoW -



	Dataset	Parameters	Accuracy	Recall	F1 Score	Precision Score
0	enron1	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.942982	0.926174	0.913907	0.901961
1	enron2	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.926778	0.792308	0.854772	0.927928
2	enron4	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.955801	1.000000	0.970223	0.942169
3	enron1	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.942982	0.926174	0.913907	0.901961
4	enron2	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.926778	0.792308	0.854772	0.927928
5	enron4	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.955801	1.000000	0.970223	0.942169
6	enron1	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.942982	0.926174	0.913907	0.901961
7	enron2	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.920502	0.769231	0.840336	0.925926
8	enron4	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.942910	1.000000	0.961870	0.926540
9	enron1	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.947368	0.932886	0.920530	0.908497
10	enron2	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.914226	0.746154	0.825532	0.923810
11	enron4	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.942910	1.000000	0.961870	0.926540

Hyperparameter tuned Bernoulli -

Name: Nahush Jayesh Patil



	Dataset	Parameters	Accuracy	Recall	F1 Score	Precision Score
0	enron1	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.945175	0.872483	0.912281	0.955882
1	enron2	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.903766	0.715385	0.801724	0.911765
2	enron4	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': False}	0.950276	1.000000	0.966625	0.935407
3	enron1	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.945175	0.872483	0.912281	0.955882
4	enron2	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.903766	0.715385	0.801724	0.911765
5	enron4	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': False}	0.950276	1.000000	0.966625	0.935407
6	enron1	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.942982	0.865772	0.908451	0.955556
7	enron2	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.887029	0.653846	0.758929	0.904255
8	enron4	{'C': 0.1, 'solver': 'liblinear', 'penalty': 'l2', 'fit_intercept': True}	0.946593	1.000000	0.964242	0.930952
9	enron1	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.936404	0.852349	0.897527	0.947761
10	enron2	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.887029	0.653846	0.758929	0.904255
11	enron4	{'C': 0.1, 'solver': 'lbfgs', 'penalty': 'l2', 'fit_intercept': True}	0.942910	1.000000	0.961870	0.926540

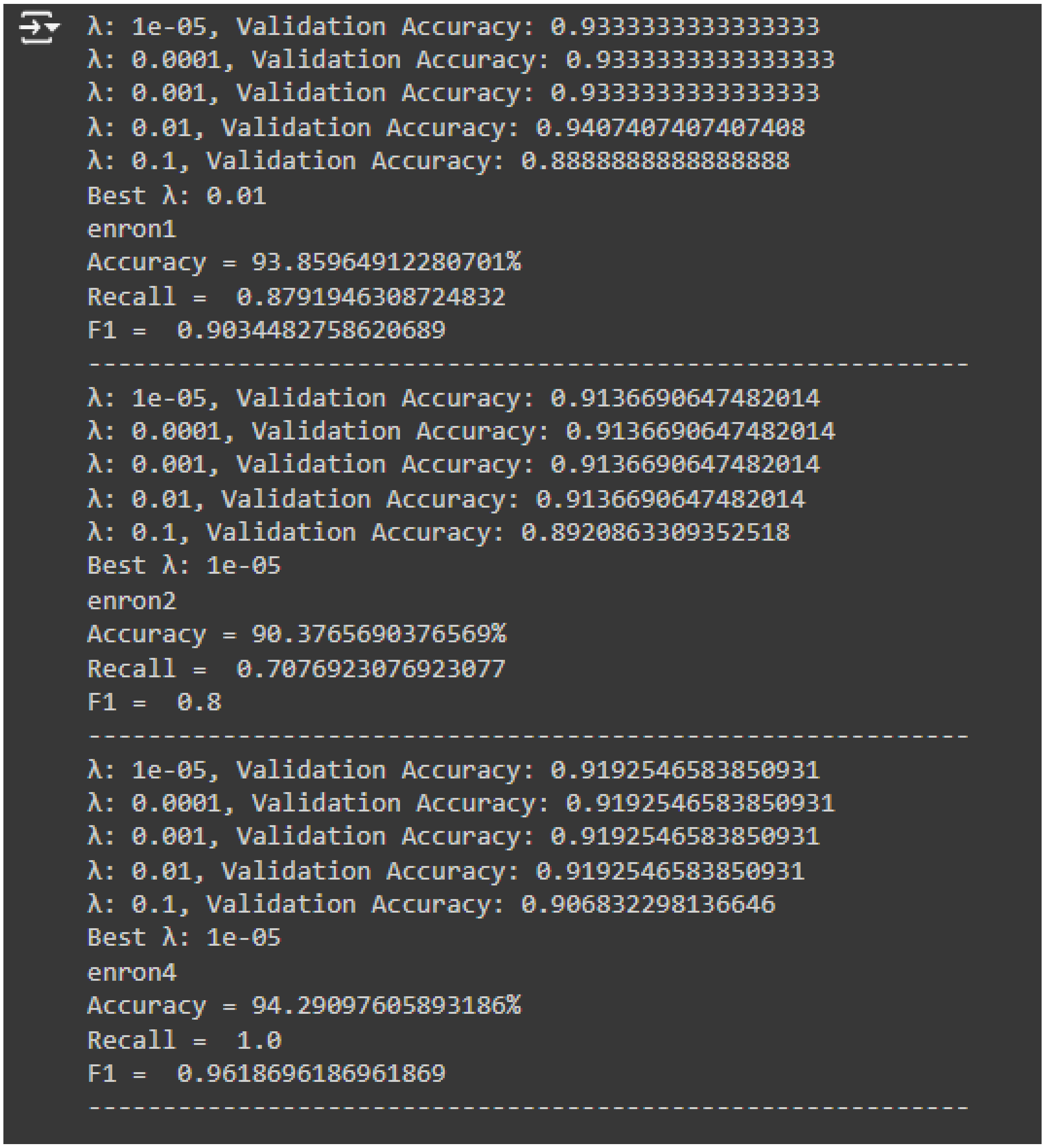
To achieve better results on the enron datasets using Logistic Regression, a hyperparameter tuning process was done. The emphasis was given on 4 hyperparameters - Regularizing constant C, Solvers, Penalty, fit_intercept

Regularizing constant C - Controls the strength of regularization. It has an inverse behavior, i.e. larger value weakens the effect of the regularizer and smaller value strengthens the effect of the regularizer.
Solvers - Algorithm used for optimization of the algorithm. The default value is 'lbfgs'. As the dataset was small, the experiment was carried out using 'liblinear' as it tends to work well with smaller datasets.
Penalty - Defines the type of penalty or regularization to use. For all the datasets, l2 regularizer worked the best.
Fit_intercept - Decides whether to add a bias term or not.

Step by step approach Bag of Words results -

Name: Nahush Jayesh Patil

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	93.85%	87.91%	90.03%	92.90%
Enron 2	90.37%	70.76%	80%	92%
Enron 4	94.29%	100%	96.18%	92.65%

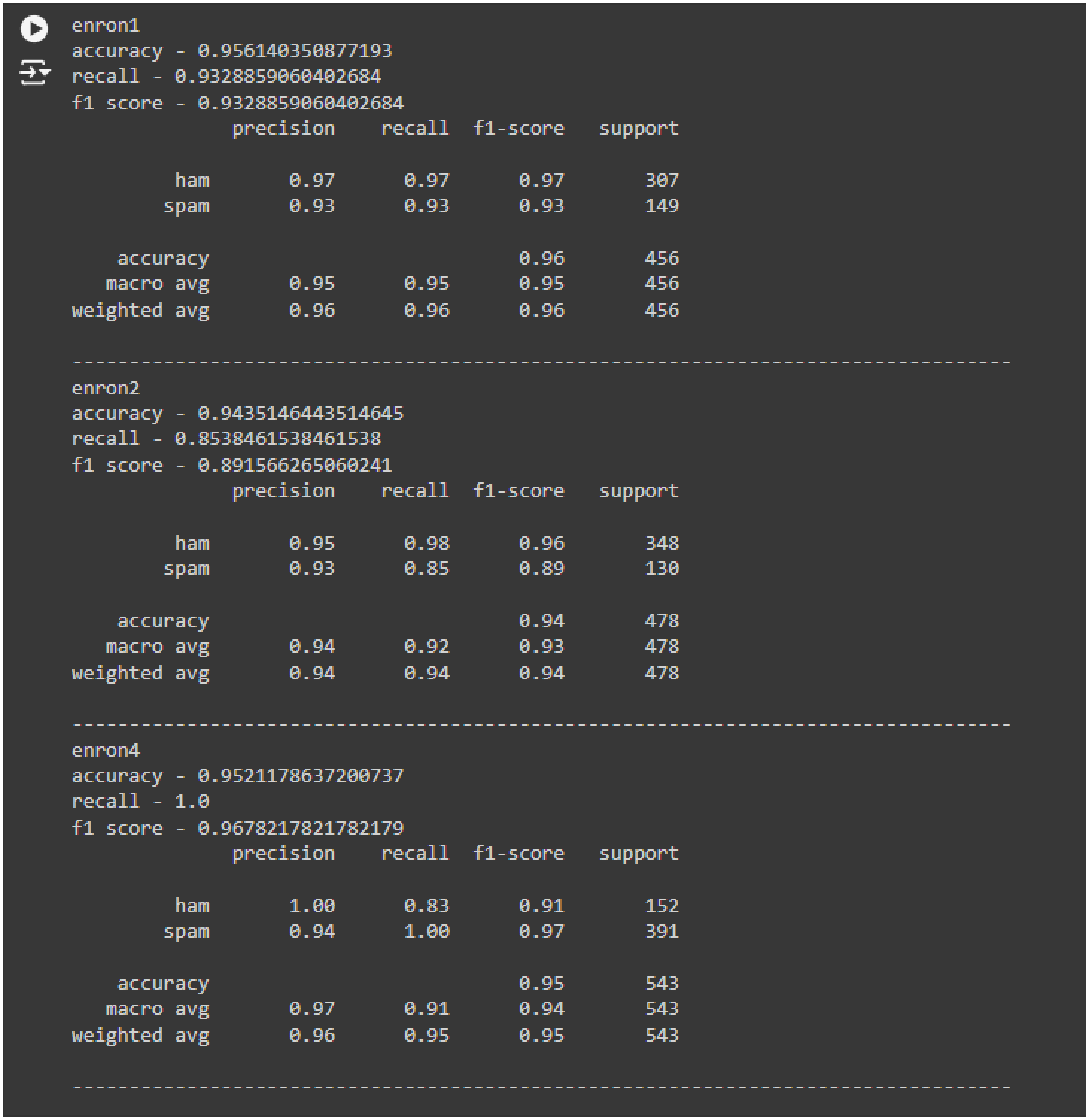


```
λ: 1e-05, Validation Accuracy: 0.9333333333333333
λ: 0.0001, Validation Accuracy: 0.9333333333333333
λ: 0.001, Validation Accuracy: 0.9333333333333333
λ: 0.01, Validation Accuracy: 0.9407407407407408
λ: 0.1, Validation Accuracy: 0.8888888888888888
Best λ: 0.01
enron1
Accuracy = 93.85964912280701%
Recall =  0.8791946308724832
F1 =  0.9034482758620689
------------------------------------------------------------
λ: 1e-05, Validation Accuracy: 0.9136690647482014
λ: 0.0001, Validation Accuracy: 0.9136690647482014
λ: 0.001, Validation Accuracy: 0.9136690647482014
λ: 0.01, Validation Accuracy: 0.9136690647482014
λ: 0.1, Validation Accuracy: 0.8920863309352518
Best λ: 1e-05
enron2
Accuracy = 90.3765690376569%
Recall =  0.7076923076923077
F1 =  0.8
------------------------------------------------------------
λ: 1e-05, Validation Accuracy: 0.9192546583850931
λ: 0.0001, Validation Accuracy: 0.9192546583850931
λ: 0.001, Validation Accuracy: 0.9192546583850931
λ: 0.01, Validation Accuracy: 0.9192546583850931
λ: 0.1, Validation Accuracy: 0.906832298136646
Best λ: 1e-05
enron4
Accuracy = 94.29097605893186%
Recall =  1.0
F1 =  0.9618696186961869
------------------------------------------------------------
```

Scikit learn approach Bernoulli results -

Name: Nahush Jayesh Patil

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	95.61%	93.28%	93.28%	97%
Enron 2	94.35%	85.38%	89.15%	97%
Enron 4	95.21%	100%	96.78%	97%

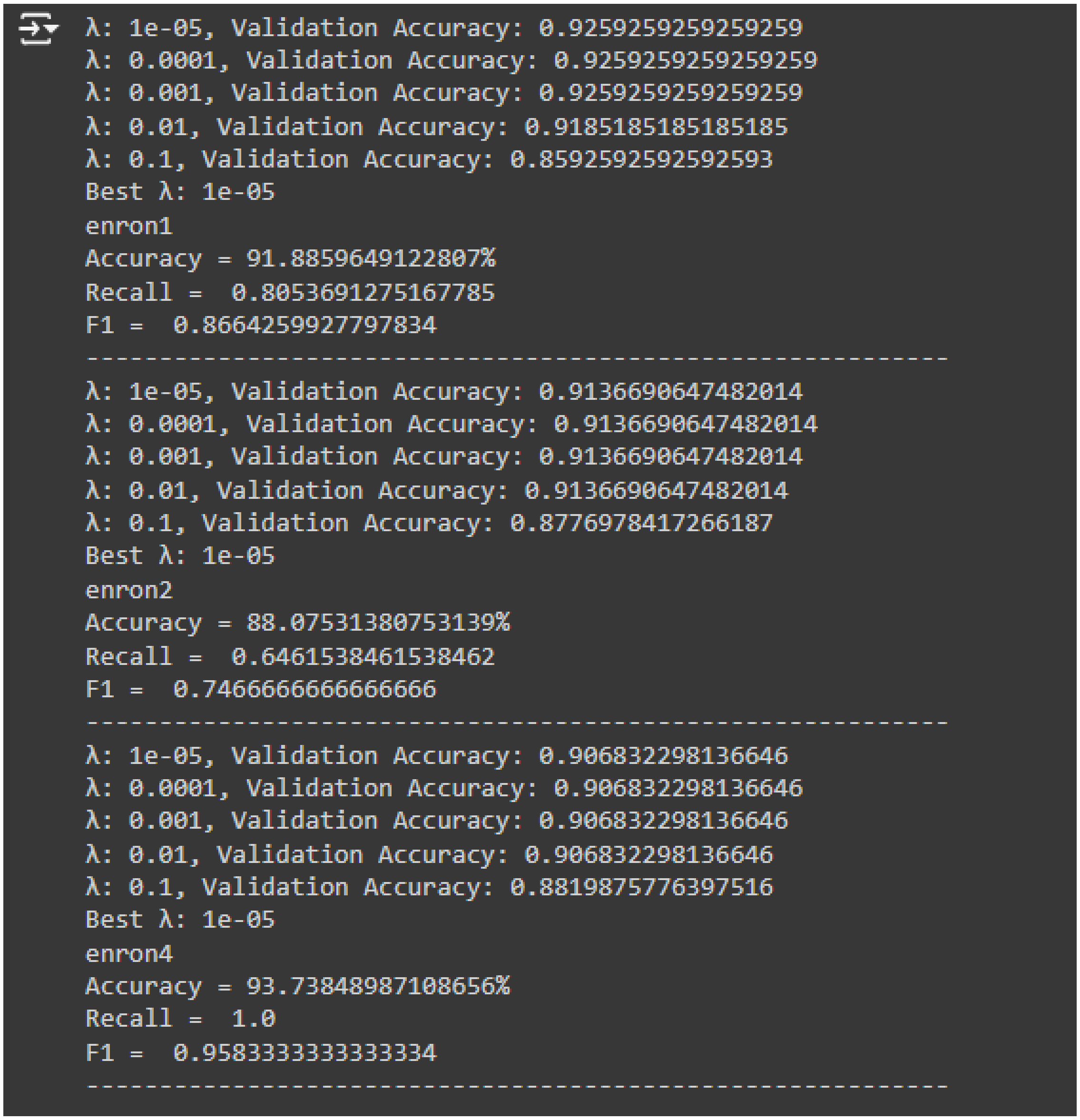


```
enron1
accuracy - 0.956140350877193
recall - 0.9328859060402684
f1 score - 0.9328859060402684
              precision    recall  f1-score   support

         ham       0.97      0.97      0.97       307
        spam       0.93      0.93      0.93       149

    accuracy                           0.96       456
   macro avg       0.95      0.95      0.95       456
weighted avg       0.96      0.96      0.96       456

---------------------------------------------------------------------------------
enron2
accuracy - 0.9435146443514645
recall - 0.8538461538461538
f1 score - 0.891566265060241
              precision    recall  f1-score   support

         ham       0.95      0.98      0.96       348
        spam       0.93      0.85      0.89       130

    accuracy                           0.94       478
   macro avg       0.94      0.92      0.93       478
weighted avg       0.94      0.94      0.94       478

---------------------------------------------------------------------------------
enron4
accuracy - 0.9521178637200737
recall - 1.0
f1 score - 0.9678217821782179
              precision    recall  f1-score   support

         ham       1.00      0.83      0.91       152
        spam       0.94      1.00      0.97       391

    accuracy                           0.95       543
   macro avg       0.97      0.91      0.94       543
weighted avg       0.96      0.95      0.95       543

---------------------------------------------------------------------------------
```

Name: Nahush Jayesh Patil

Step by step approach results -

Dataset	Accuracy	Recall score	F1 score
Enron 1	91.88%	80.53%	93.75%
Enron 2	88.07%	64.61%	88.4%
Enron 4	93.73%	100%	92%



```
λ: 1e-05, Validation Accuracy: 0.9259259259259259
λ: 0.0001, Validation Accuracy: 0.9259259259259259
λ: 0.001, Validation Accuracy: 0.9259259259259259
λ: 0.01, Validation Accuracy: 0.9185185185185185
λ: 0.1, Validation Accuracy: 0.8592592592592593
Best λ: 1e-05
enron1
Accuracy = 91.8859649122807%
Recall =  0.8053691275167785
F1 =  0.8664259927797834
------------------------------------------------------------
λ: 1e-05, Validation Accuracy: 0.9136690647482014
λ: 0.0001, Validation Accuracy: 0.9136690647482014
λ: 0.001, Validation Accuracy: 0.9136690647482014
λ: 0.01, Validation Accuracy: 0.9136690647482014
λ: 0.1, Validation Accuracy: 0.8776978417266187
Best λ: 1e-05
enron2
Accuracy = 88.07531380753139%
Recall =  0.6461538461538462
F1 =  0.7466666666666666
------------------------------------------------------------
λ: 1e-05, Validation Accuracy: 0.906832298136646
λ: 0.0001, Validation Accuracy: 0.906832298136646
λ: 0.001, Validation Accuracy: 0.906832298136646
λ: 0.01, Validation Accuracy: 0.906832298136646
λ: 0.1, Validation Accuracy: 0.8819875776397516
Best λ: 1e-05
enron4
Accuracy = 93.73848987108656%
Recall =  1.0
F1 =  0.9583333333333334
------------------------------------------------------------
```

STOCHASTIC GRADIENT DESCENT

Name: Nahush Jayesh Patil

Scikit learn Bag of Words approach -

For enron 1 - Best Parameters: {'alpha': 0.001, 'early_stopping': False, 'learning_rate': 'optimal', 'loss': 'modified_huber', 'max_iter': 1000, 'penalty': None, 'validation_fraction': 0.2, 'warm_start': False}

For enron 2 - Best Parameters: {'alpha': 0.01, 'early_stopping': False, 'learning_rate': 'optimal', 'loss': 'hinge', 'max_iter': 800, 'penalty': 'l2', 'validation_fraction': 0.3, 'warm_start': True}

For enron 4 - Best Parameters: {'alpha': 0.0001, 'early_stopping': False, 'learning_rate': 'optimal', 'loss': 'perceptron', 'max_iter': 1000, 'penalty': None, 'validation_fraction': 0.3, 'warm_start': True}

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	91.22%	80.91%	86.48%	93.75%
Enron 2	94.97%	93.07%	90.97%	88.42%
Enron 4	95.76%	96.93%	97.05%	92%

Scikit learn Bernoulli approach -

Dataset	Accuracy	Recall score	F1 score	Precision score
Enron 1	97.07%	96.92%	94.93%	93.75%
Enron 2	94.97%	93.07%	90.97%	88.42%
Enron 4	96.66%	98.20%	97.77%	92%

POST FEATURE ENGINEERING -

Name: Nahush Jayesh Patil

To improve the performance of the models, the following features were added -

- Word count
- Average word length
- Number of nouns
- Number of adjectives
- Number of verbs
- Number of special characters
- Number of numeric characters

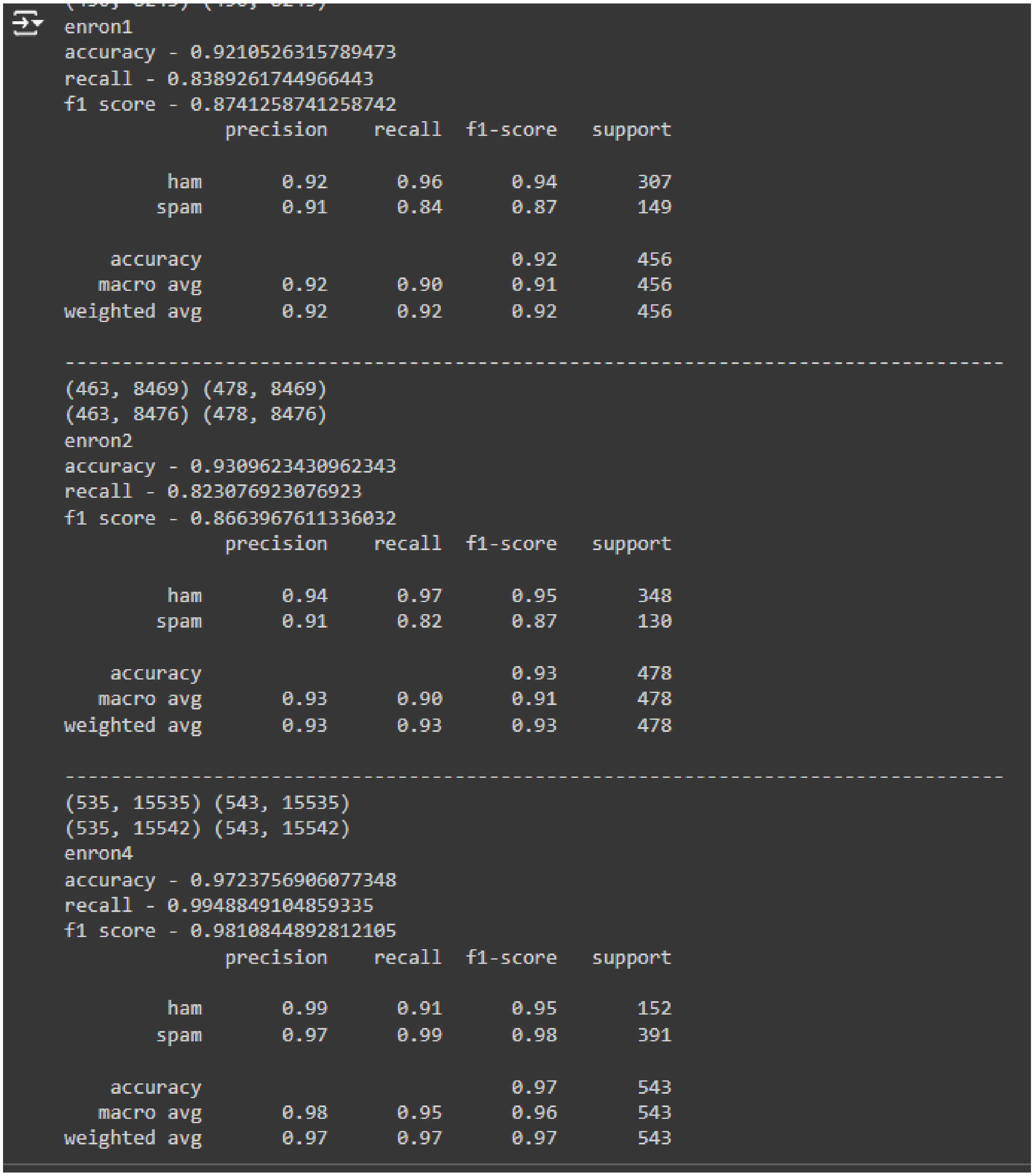
These continuous variables were first discretized using binning. Bins were created based on quantiles and divided into 4 groups - Low, Medium, High and Very High. These categorical variables were then encoded using Scikit Learn's Label encoder.

RESULTS

1. **Multinomial Naive Bayes (BoW)**

Dataset	Accuracy	Recall score	F1 score
Enron 1	92.10%	83.89%	87.41%
Enron 2	93.10%	82.30%	86.63%
Enron 4	97.23%	99.48%	98.10%

Name: Nahush Jayesh Patil



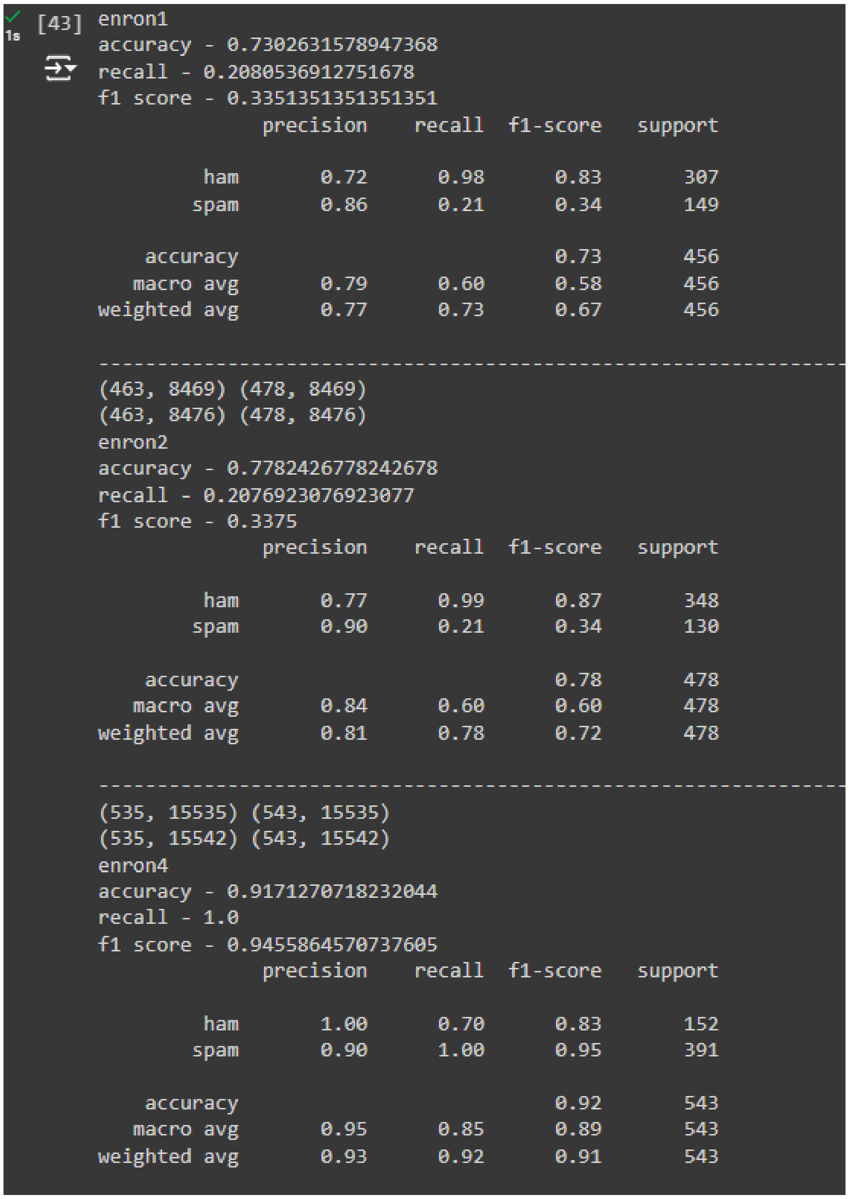
```
enron1
accuracy - 0.9210526315789473
recall - 0.8389261744966443
f1 score - 0.8741258741258742
              precision    recall  f1-score   support

         ham       0.92      0.96      0.94       307
        spam       0.91      0.84      0.87       149

    accuracy                           0.92       456
   macro avg       0.92      0.90      0.91       456
weighted avg       0.92      0.92      0.92       456

----------------------------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.9309623430962343
recall - 0.823076923076923
f1 score - 0.8663967611336032
              precision    recall  f1-score   support

         ham       0.94      0.97      0.95       348
        spam       0.91      0.82      0.87       130

    accuracy                           0.93       478
   macro avg       0.93      0.90      0.91       478
weighted avg       0.93      0.93      0.93       478

----------------------------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9723756906077348
recall - 0.9948849104859335
f1 score - 0.9810844892812105
              precision    recall  f1-score   support

         ham       0.99      0.91      0.95       152
        spam       0.97      0.99      0.98       391

    accuracy                           0.97       543
   macro avg       0.98      0.95      0.96       543
weighted avg       0.97      0.97      0.97       543
```

Observation - Slight decrease in performance

Name: Nahush Jayesh Patil

2. Multinomial Naive Bayes (Bernoulli)

Dataset	Accuracy	Recall score	F1 score
Enron 1	73.02%	20.80%	33.51%
Enron 2	77.82%	20.76%	33.75%
Enron 4	91.71%	100%	94.55%



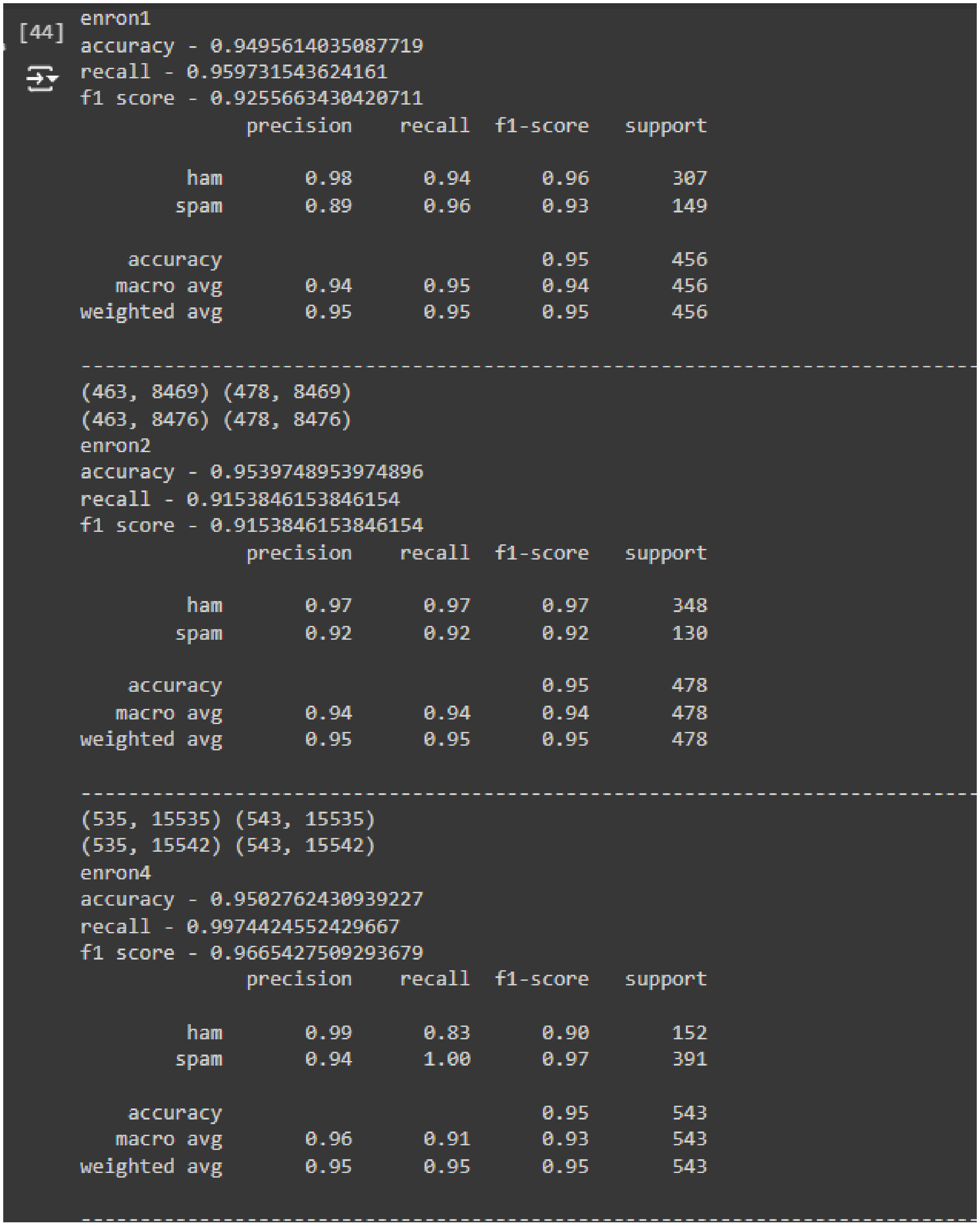
```
enron1
accuracy - 0.7302631578947368
recall - 0.2080536912751678
f1 score - 0.3351351351351351
              precision    recall  f1-score   support

         ham       0.72      0.98      0.83       307
        spam       0.86      0.21      0.34       149

    accuracy                           0.73       456
   macro avg       0.79      0.60      0.58       456
weighted avg       0.77      0.73      0.67       456

----------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.7782426778242678
recall - 0.2076923076923077
f1 score - 0.3375
              precision    recall  f1-score   support

         ham       0.77      0.99      0.87       348
        spam       0.90      0.21      0.34       130

    accuracy                           0.78       478
   macro avg       0.84      0.60      0.60       478
weighted avg       0.81      0.78      0.72       478

----------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9171270718232044
recall - 1.0
f1 score - 0.9455864570737605
              precision    recall  f1-score   support

         ham       1.00      0.70      0.83       152
        spam       0.90      1.00      0.95       391

    accuracy                           0.92       543
   macro avg       0.95      0.85      0.89       543
weighted avg       0.93      0.92      0.91       543
```

Observation - No change in the results

Name: Nahush Jayesh Patil

3. LR Bag of Words

Dataset	Accuracy	Recall score	F1 score
Enron 1	94.95%	95.97%	92.55%
Enron 2	95.39%	91.53%	91.53%
Enron 4	95.02%	99.74%	96.65%



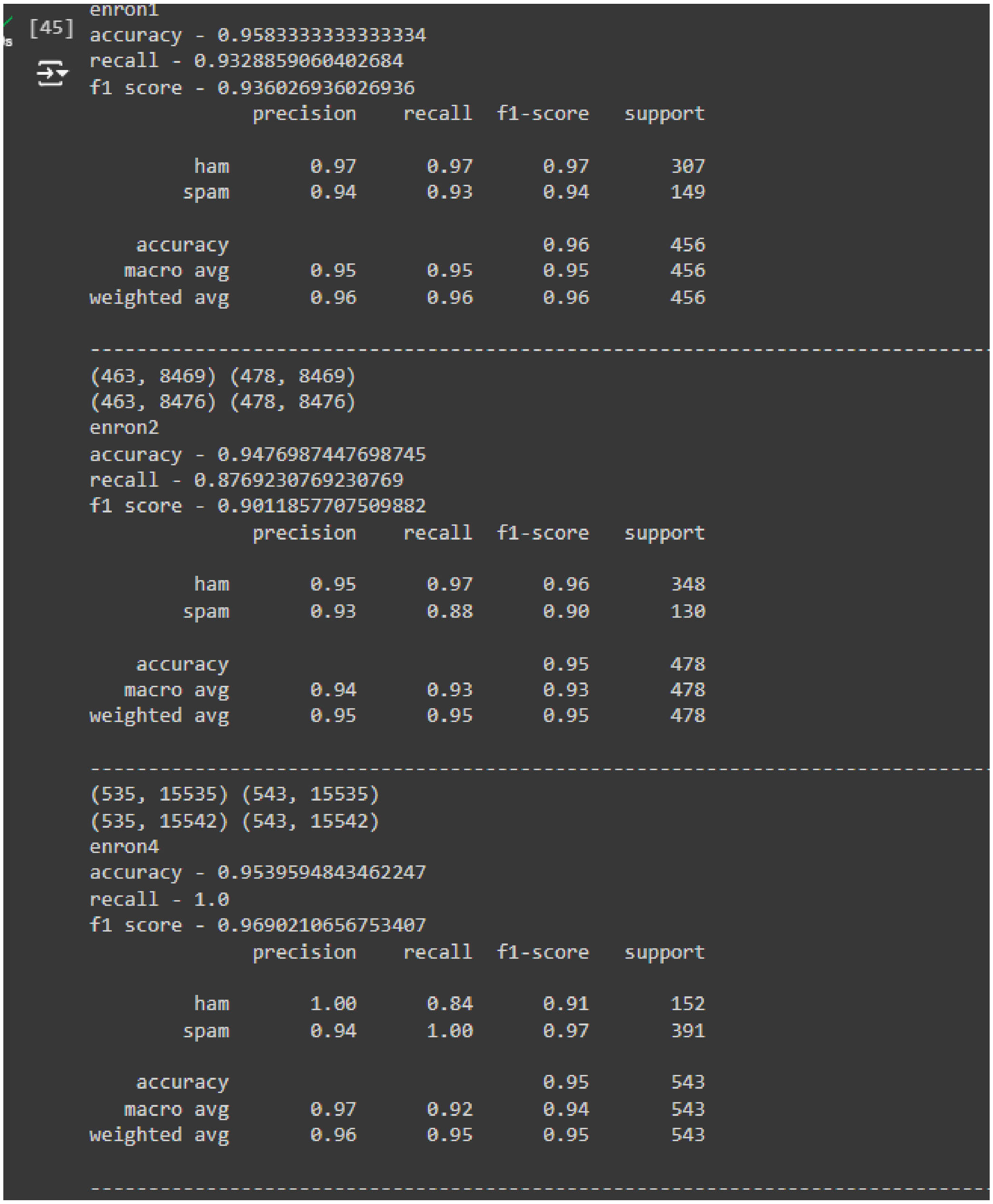
```
enron1
accuracy - 0.9495614035087719
recall - 0.959731543624161
f1 score - 0.9255663430420711
              precision    recall  f1-score   support

         ham       0.98      0.94      0.96       307
        spam       0.89      0.96      0.93       149

    accuracy                           0.95       456
   macro avg       0.94      0.95      0.94       456
weighted avg       0.95      0.95      0.95       456

--------------------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.9539748953974896
recall - 0.9153846153846154
f1 score - 0.9153846153846154
              precision    recall  f1-score   support

         ham       0.97      0.97      0.97       348
        spam       0.92      0.92      0.92       130

    accuracy                           0.95       478
   macro avg       0.94      0.94      0.94       478
weighted avg       0.95      0.95      0.95       478

--------------------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9502762430939227
recall - 0.9974424552429667
f1 score - 0.9665427509293679
              precision    recall  f1-score   support

         ham       0.99      0.83      0.90       152
        spam       0.94      1.00      0.97       391

    accuracy                           0.95       543
   macro avg       0.96      0.91      0.93       543
weighted avg       0.95      0.95      0.95       543

--------------------------------------------------------------------------
```

Observation - Slight decrease in performance

4. LR using Bernoulli

Dataset	Accuracy	Recall score	F1 score
Enron 1	95.83%	93.28%	93.60%
Enron 2	94.76%	87.69%	90.11%
Enron 4	95.40%	100%	96.90%

Name: Nahush Jayesh Patil



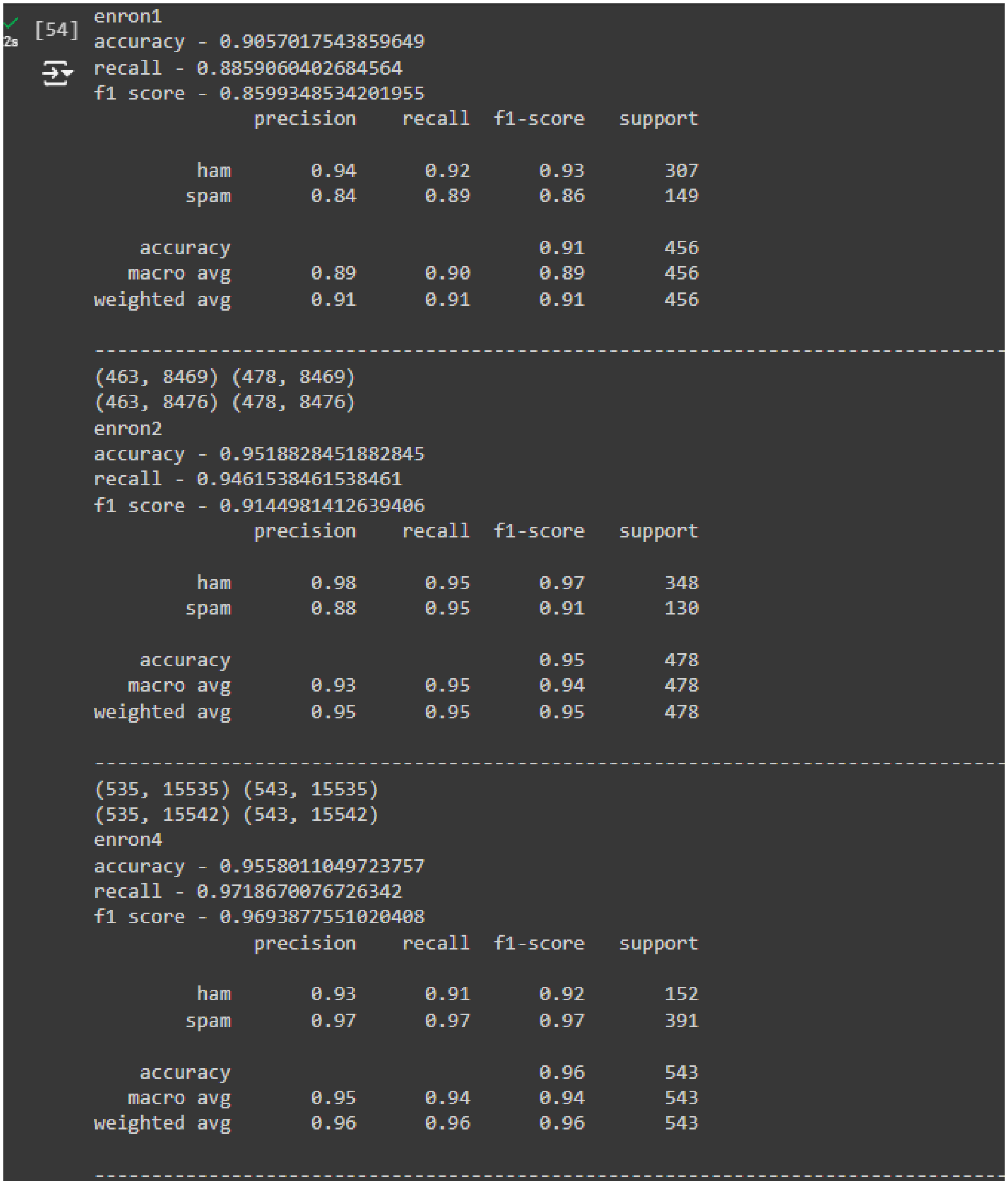
```
enron1
accuracy - 0.9583333333333334
recall - 0.9328859060402684
f1 score - 0.936026936026936
              precision    recall  f1-score   support

         ham       0.97      0.97      0.97       307
        spam       0.94      0.93      0.94       149

    accuracy                           0.96       456
   macro avg       0.95      0.95      0.95       456
weighted avg       0.96      0.96      0.96       456

-----------------------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.9476987447698745
recall - 0.8769230769230769
f1 score - 0.9011857707509882
              precision    recall  f1-score   support

         ham       0.95      0.97      0.96       348
        spam       0.93      0.88      0.90       130

    accuracy                           0.95       478
   macro avg       0.94      0.93      0.93       478
weighted avg       0.95      0.95      0.95       478

-----------------------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9539594843462247
recall - 1.0
f1 score - 0.9690210656753407
              precision    recall  f1-score   support

         ham       1.00      0.84      0.91       152
        spam       0.94      1.00      0.97       391

    accuracy                           0.95       543
   macro avg       0.97      0.92      0.94       543
weighted avg       0.96      0.95      0.95       543

-----------------------------------------------------------------------------
```

Observation - Slight increase in the performance

Name: Nahush Jayesh Patil

5. SGDClassifier BoW

Dataset	Accuracy	Recall score	F1 score
Enron 1	90.57%	88.59%	85.99%
Enron 2	95.18%	94.61%	91.44%
Enron 4	95.58%	97.18%	96.93%



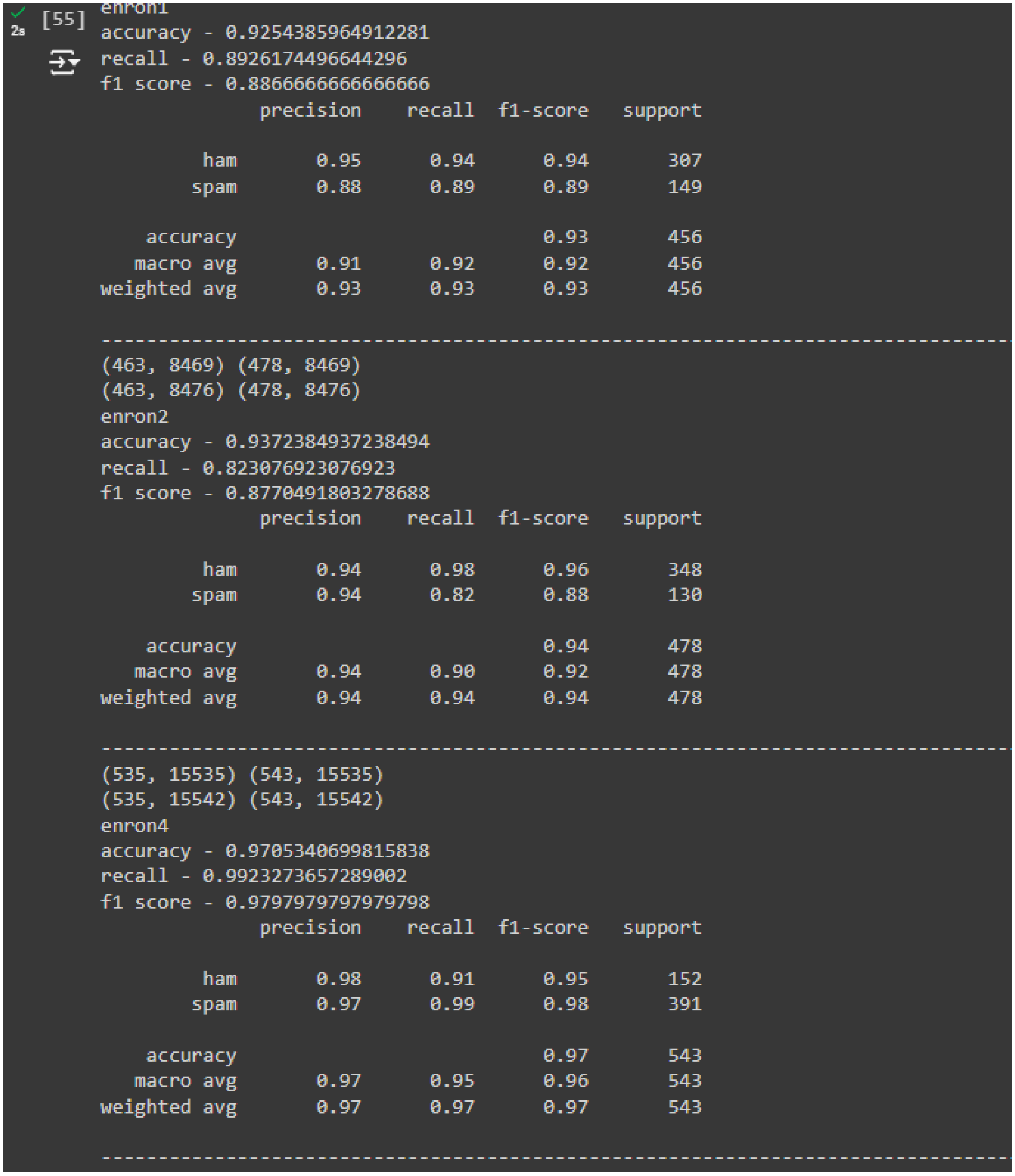
```
enron1
accuracy - 0.9057017543859649
recall - 0.8859060402684564
f1 score - 0.8599348534201955
              precision    recall  f1-score   support

         ham       0.94      0.92      0.93       307
        spam       0.84      0.89      0.86       149

    accuracy                           0.91       456
   macro avg       0.89      0.90      0.89       456
weighted avg       0.91      0.91      0.91       456

-------------------------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.9518828451882845
recall - 0.9461538461538461
f1 score - 0.9144981412639406
              precision    recall  f1-score   support

         ham       0.98      0.95      0.97       348
        spam       0.88      0.95      0.91       130

    accuracy                           0.95       478
   macro avg       0.93      0.95      0.94       478
weighted avg       0.95      0.95      0.95       478

-------------------------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9558011049723757
recall - 0.9718670076726342
f1 score - 0.9693877551020408
              precision    recall  f1-score   support

         ham       0.93      0.91      0.92       152
        spam       0.97      0.97      0.97       391

    accuracy                           0.96       543
   macro avg       0.95      0.94      0.94       543
weighted avg       0.96      0.96      0.96       543

-------------------------------------------------------------------------------
```

Observation - For enron 1 and 4 recall score slightly increased. For enron2 there was slight improvement in all the metrics.

Name: Nahush Jayesh Patil

6. SGDClassifier Bernoulli -

Dataset	Accuracy	Recall score	F1 score
Enron 1	92.54%	89.26%	88.66%
Enron 2	93.72%	82.30%	87.70%
Enron 4	97.05%	99.23%	97.97%



```
enron1
accuracy - 0.9254385964912281
recall - 0.8926174496644296
f1 score - 0.8866666666666666
              precision    recall  f1-score   support

         ham       0.95      0.94      0.94       307
        spam       0.88      0.89      0.89       149

    accuracy                           0.93       456
   macro avg       0.91      0.92      0.92       456
weighted avg       0.93      0.93      0.93       456

---------------------------------------------------------------------------------
(463, 8469) (478, 8469)
(463, 8476) (478, 8476)
enron2
accuracy - 0.9372384937238494
recall - 0.823076923076923
f1 score - 0.8770491803278688
              precision    recall  f1-score   support

         ham       0.94      0.98      0.96       348
        spam       0.94      0.82      0.88       130

    accuracy                           0.94       478
   macro avg       0.94      0.90      0.92       478
weighted avg       0.94      0.94      0.94       478

---------------------------------------------------------------------------------
(535, 15535) (543, 15535)
(535, 15542) (543, 15542)
enron4
accuracy - 0.9705340699815838
recall - 0.9923273657289002
f1 score - 0.9797979797979798
              precision    recall  f1-score   support

         ham       0.98      0.91      0.95       152
        spam       0.97      0.99      0.98       391

    accuracy                           0.97       543
   macro avg       0.97      0.95      0.96       543
weighted avg       0.97      0.97      0.97       543

---------------------------------------------------------------------------------
```

Name: Nahush Jayesh Patil

Observation - Slight increase in performance for enron4 but significant decrease in performance for enron 1 and 2.

Answer the following questions:

1. Which data representation and algorithm combination yields the best performance (measured in terms of the accuracy, precision, recall and F1 score) and why?

For accuracy, precision and f1 score, the Bag of Words (BoW) representation with Multinomial Naive Bayes yields the best performance. However, in terms of Recall score the Bernoulli representation with the Discrete Naive Bayes gives the best results.

BoW tends to work better than the Bernoulli representation because it captures the frequency of the token in the document, providing more information about the importance of the word. The idea here is that the higher the frequency of the token the more is the importance. Bernoulli representation, on the other hand, only indicates the presence of a token which is not enough to capture the context of data to classify a document.

The dataset provided is small in size due to which Multinomial Naive Bayes works better because of its generative nature. It takes into consideration the prior probabilities which is useful in case of smaller datasets. In contrast, Logistic Regression generally requires larger datasets to perform optimally, as it is a discriminative model that needs sufficient data to learn complex decision boundaries.

2. Does Multinomial Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bag of words representation? Explain your yes/no answer.

Multinomial Naive Bayes (MNB) works generally better than LR and SGDClassifier while working with Bag of Words. MNB tends to work well with discrete values which fits with the BoW representation.While MNB assumes independence between features (a strong assumption) it is suitable for higher dimensional data. LR and SGDClassifier assume a linear relationship to draw a decision boundary between the classes which may not be ideal for the BoW representation. MNB also takes into consideration the prior probabilities which is useful when there is not a lot of data, hence it works well with smaller datasets as compared to Logistic Regression which needs large amounts of data.

3. Does Discrete Naive Bayes perform better (again performance is measured in terms of the accuracy, precision, recall and F1 score) than LR and SGDClassifier on the Bernoulli representation? Explain your yes/no answer.

Yes, Discrete Naive Bayes (DNB) performs better than the LR and SGDClassifier on the Bernoulli Representation. DNB assumes independence between the data points which fits well with the Bernoulli representation of the data. The Bernoulli representation indicates the presence of a word in the document. While the assumption of independence is generally strong (and rarely holds true in real-world scenarios), it tends to work effectively in high-dimensional data, such as text classification, where feature dependencies may not be as critical. DNB can handle sparse datasets better than LR and SGD. The Bernoulli representation provides sparse data as most of the words are absent in a given document. LR and SGDClassifier assume a linear relationship between features and class labels which may not be suitable for the Bernoulli representation of the data.

4. Does your LR implementation outperform the SGDClassifier (again performance is measured in terms of the accuracy, precision, recall and F1 score) or is the difference in
performance minor? Explain your yes/no answer.

Yes LR outperforms SGDClassfier. The Logistic Regression algorithm uses Gradient Ascent for updating the weights where the entire dataset is used for calculating the gradients. On the other hand, the SGD Classifier uses Stochastic Gradient Descent which arbitrarily selects a point from the dataset to calculate the gradient. Due to this behavior, SGD is computationally efficient and converges faster compared to LR but introduces a lot of noise in the calculation of gradients. LR is smooth with its convergence and does not introduce a lot of errors while updating the gradients.