Machine Learning

Offensive Tweet Detection

November 16, 2018

Diogo Daniel Ferreira *76504*

Luís Davide Leira *76514*

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

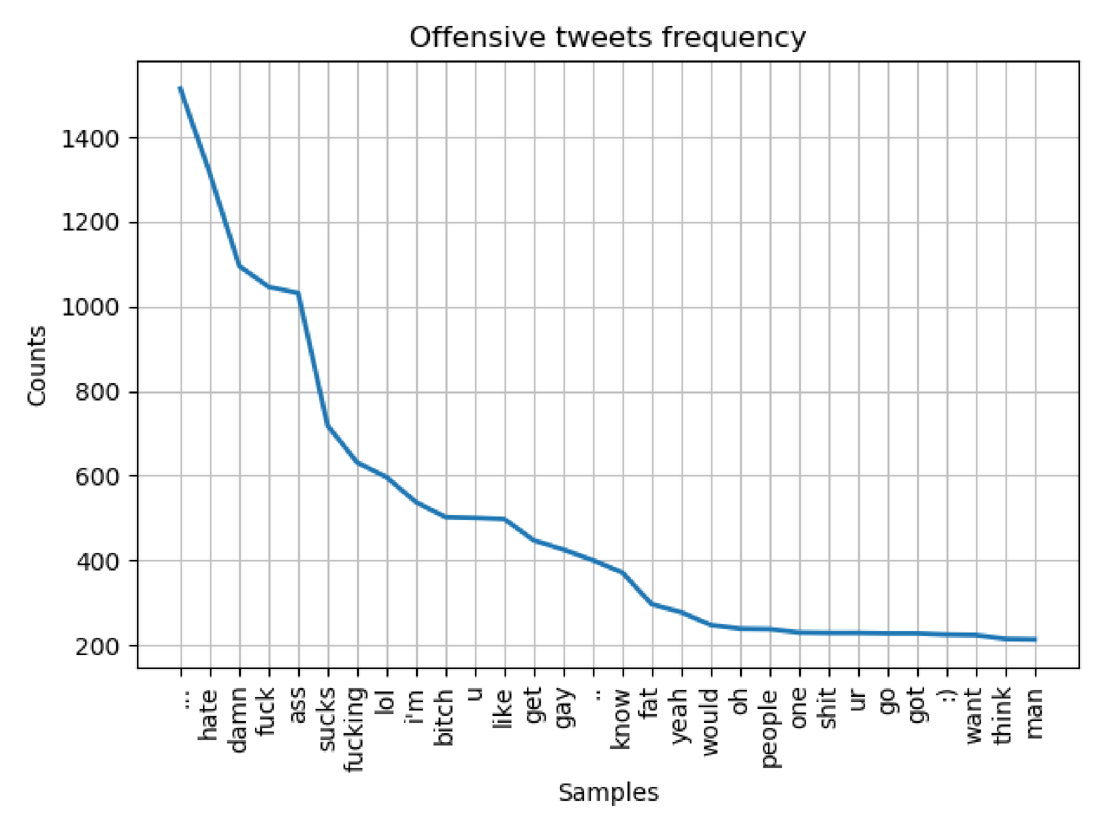
Problem

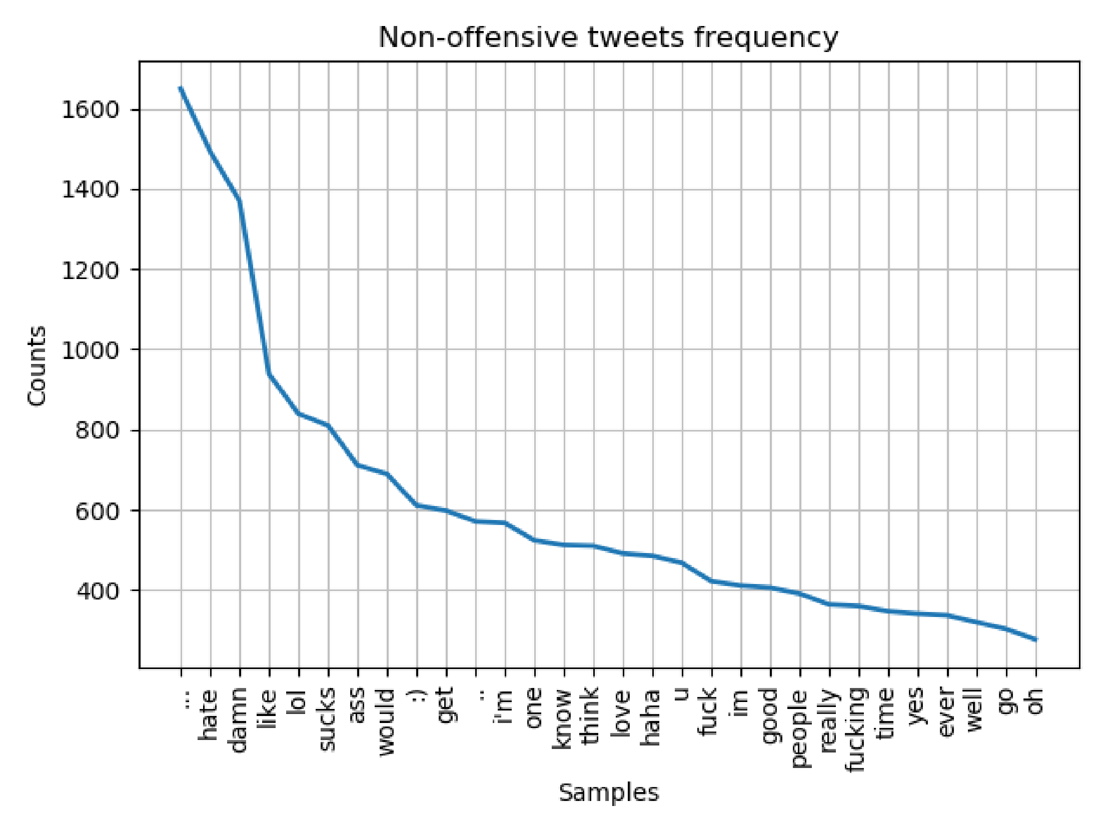
- Improve the Twitter implementation of “Safe Search”
- Don’t show offensive tweets from users with “safe search” on
- Also useful when large websites use Twitter widget to show tweets related with a topic
- Twitter can suspend or delete accounts with successive violations of safe tweets

Dataset

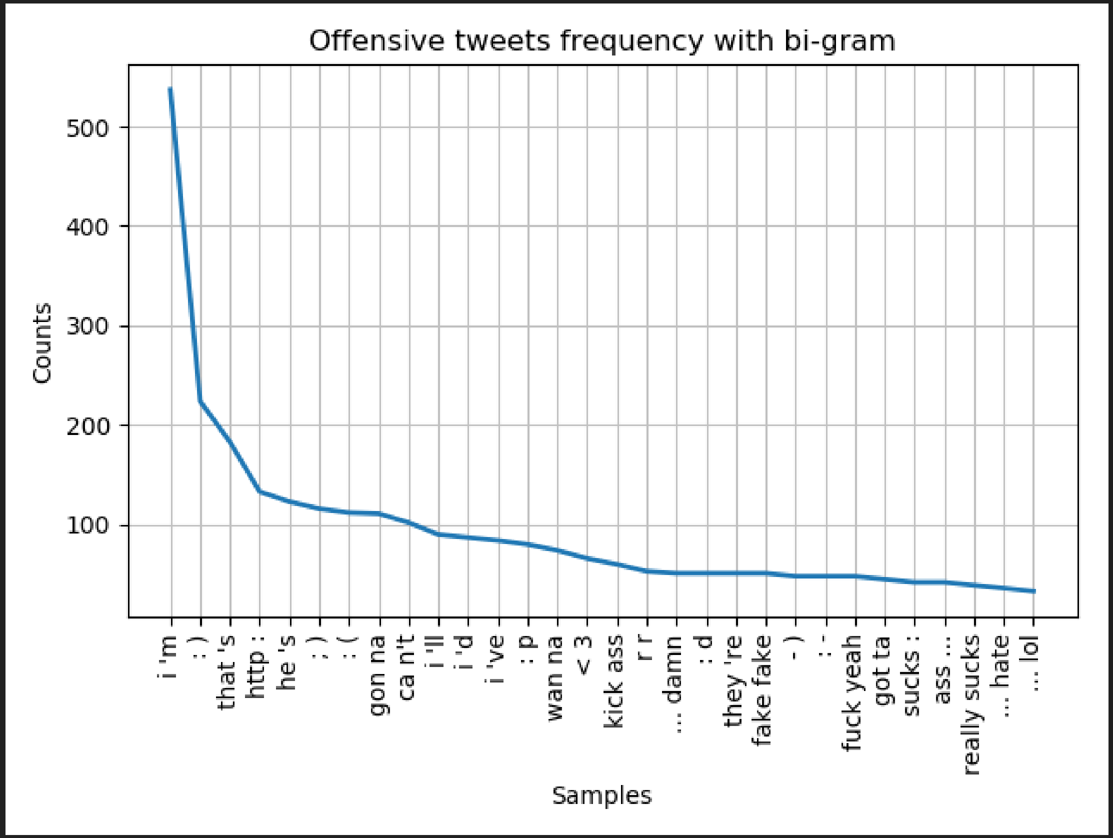
- 20001 tweets
- Human-labelled data in two categories:
  - 1 (offensive)
  - 0 (non offensive)
- 7822 labelled as offensive tweets and 12179 labelled as non offensive tweets
- Source: https://www.kaggle.com/datatlurks/dataset-for-detection-of-cybertrolls/

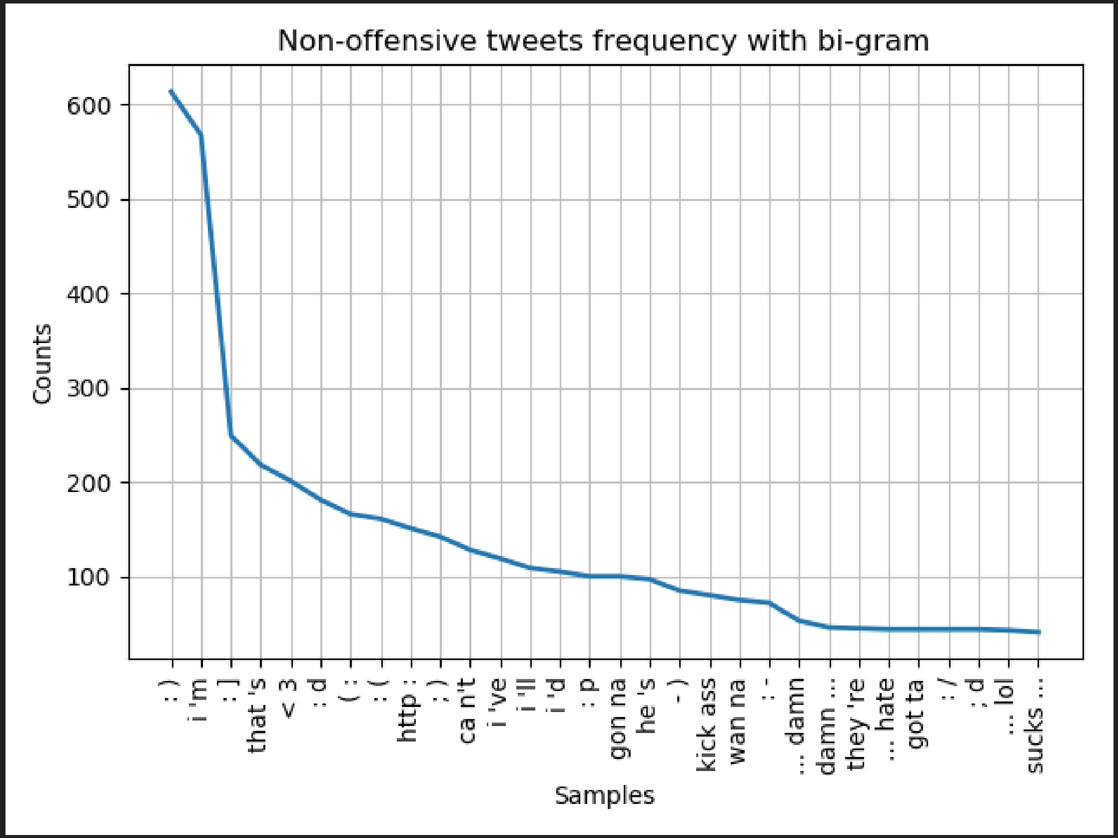
Dataset statistics





Dataset statistics





Data pre-processing

- Convert all words into lowercase
- Apply tokenizer
- Remove punctuation
- Remove stop words (the, and, or, …)
- Lemmatize

Feature extraction

Tokens counting

- Select the N most used tokens and transform them into features
- Count the number of feature tokens that each tweet has
- Normalize the counting

The pre-processed tokens can be grouped in a contiguous sequence: n-gram

We have chosen to treat each one as a word. Example:

- 1-gram: [“We”, “Love”, “AI”]
- 2-gram: [“We Love”, “Love AI”]

Feature extraction

TF-IDF weighting

- Select the N most used tokens and transform them into features
- Calculate the TF-IDF for the feature tokens that each tweet has
  - TF-IDF is a numerical statistic that reflects the importance of a token in a tweet
    - TF – term frequency
    - IDF – inverse document frequency

Classifiers

Initially, 11 classifiers were put to test:

- Logistic Regression
- SGD Classifier
- Linear SVC
- NuSVC
- SVC
- Gaussian Naive Bayes
- Decision Tree
- Random Forest
- AdaBoost
- XGBoost
- Multi-layer Perceptrons

Training & Testing approach

Dataset division: 80% train and cross-validation (5-fold cross-validation), 20% test

Parameters to be taken into consideration:

- Maximum number of tokens to create features
- Different N-gram parameters (for the tokens counting)
- Token counting and TF-IDF weighting combination
- Classifiers hyperparameters

Too many combinations!

Training & Testing approach

First phase:

- Test all classifiers with the default hyperparameters
- Test with tokens counting, TF-IDF and both combined
- Test with the following number of most used tokens:
  - 100, 1000, 5000, 10000, 25000
- Test with 1-gram, 2-gram and both combined (only applied to tokens counting)
- Perform 5-fold validation and store the confusion matrix

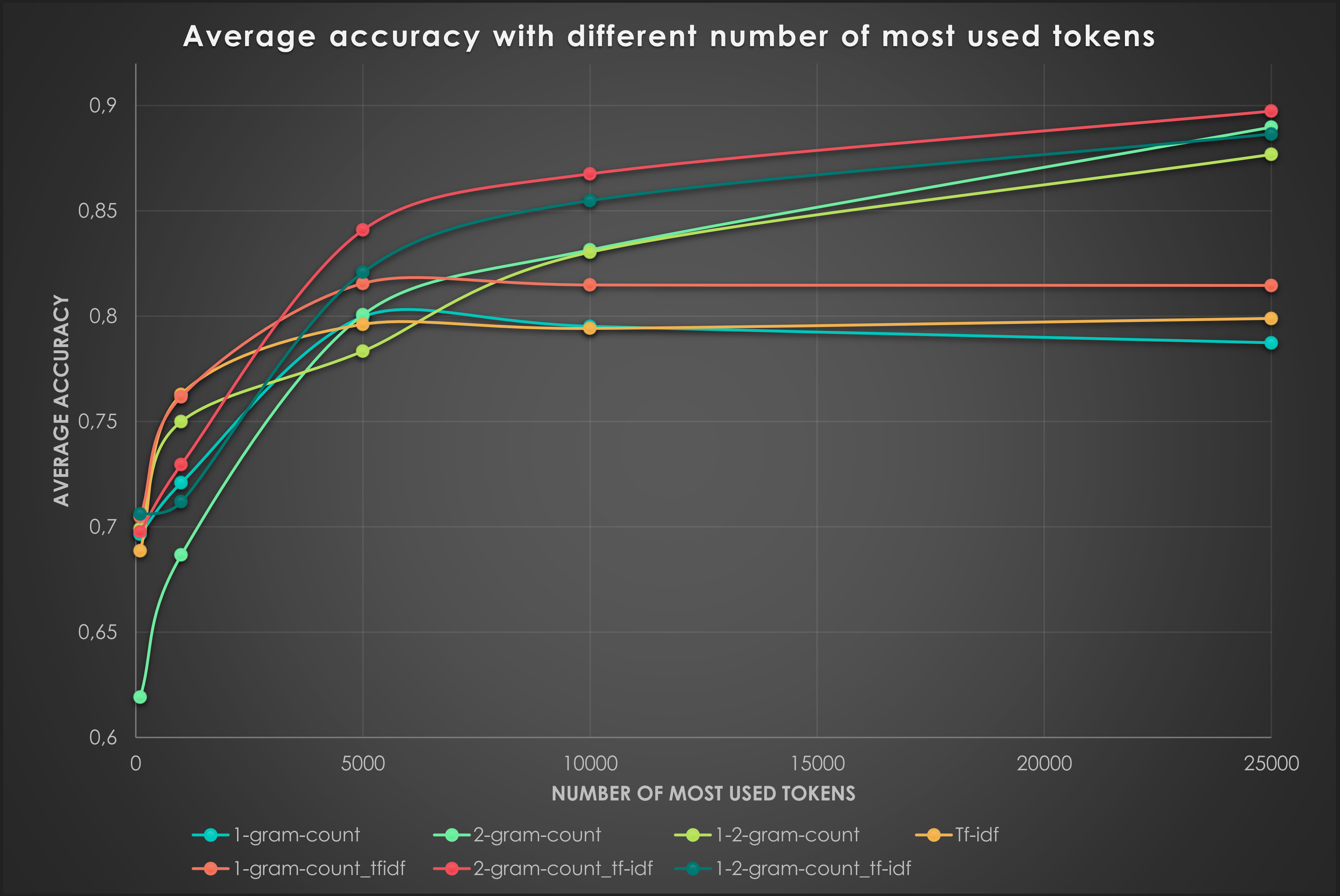
Training & Testing approach

Second phase:

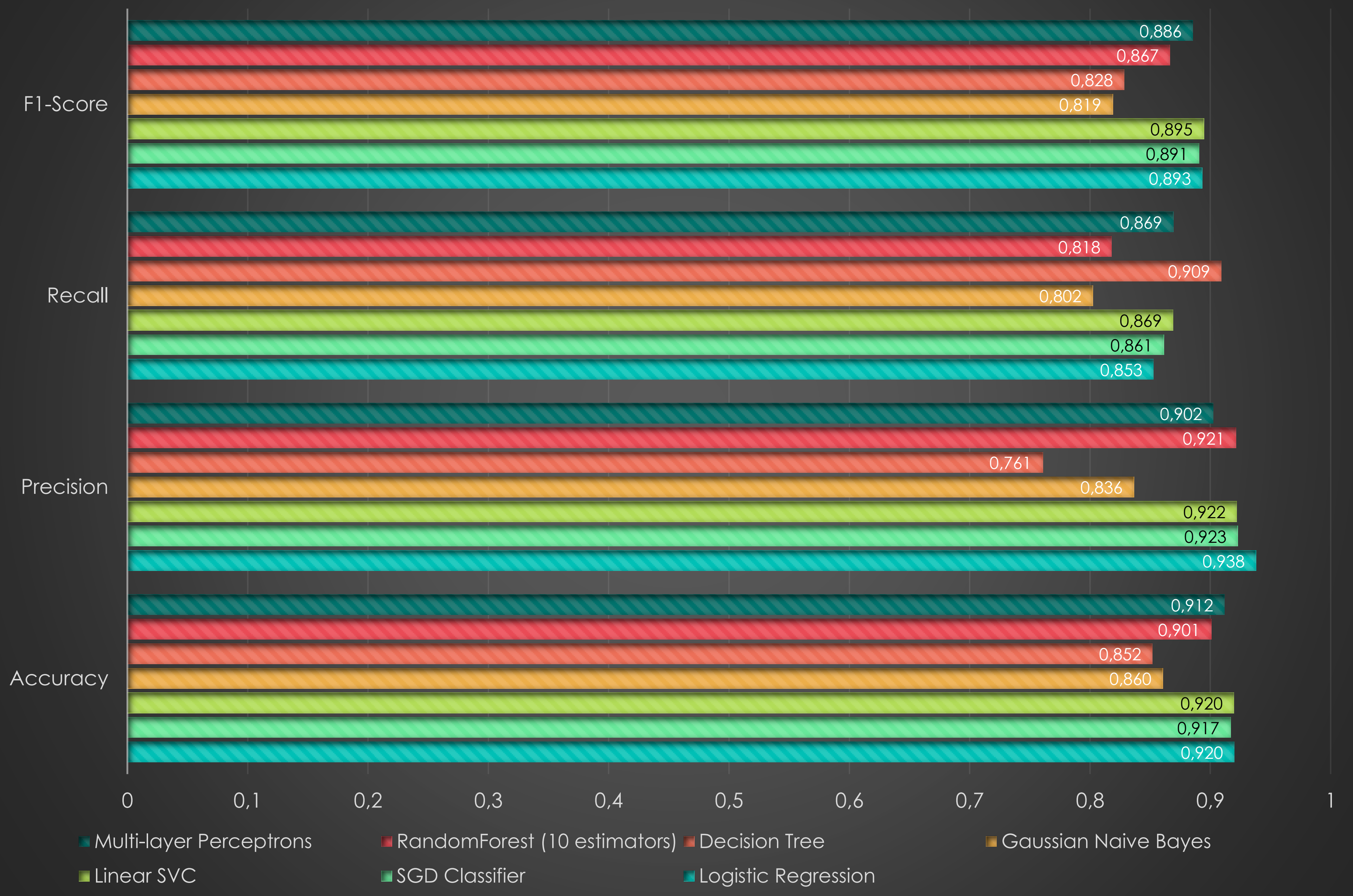
- Use the four best classifiers from the first phase
- Perform 5-fold cross validation on those four classifiers with different hyperparameters
- Select the best hyperparameters for each one of them and get the confusion matrix on the test data

Third phase:

- Test the four best classifiers with the best hyperparameters from the second phase

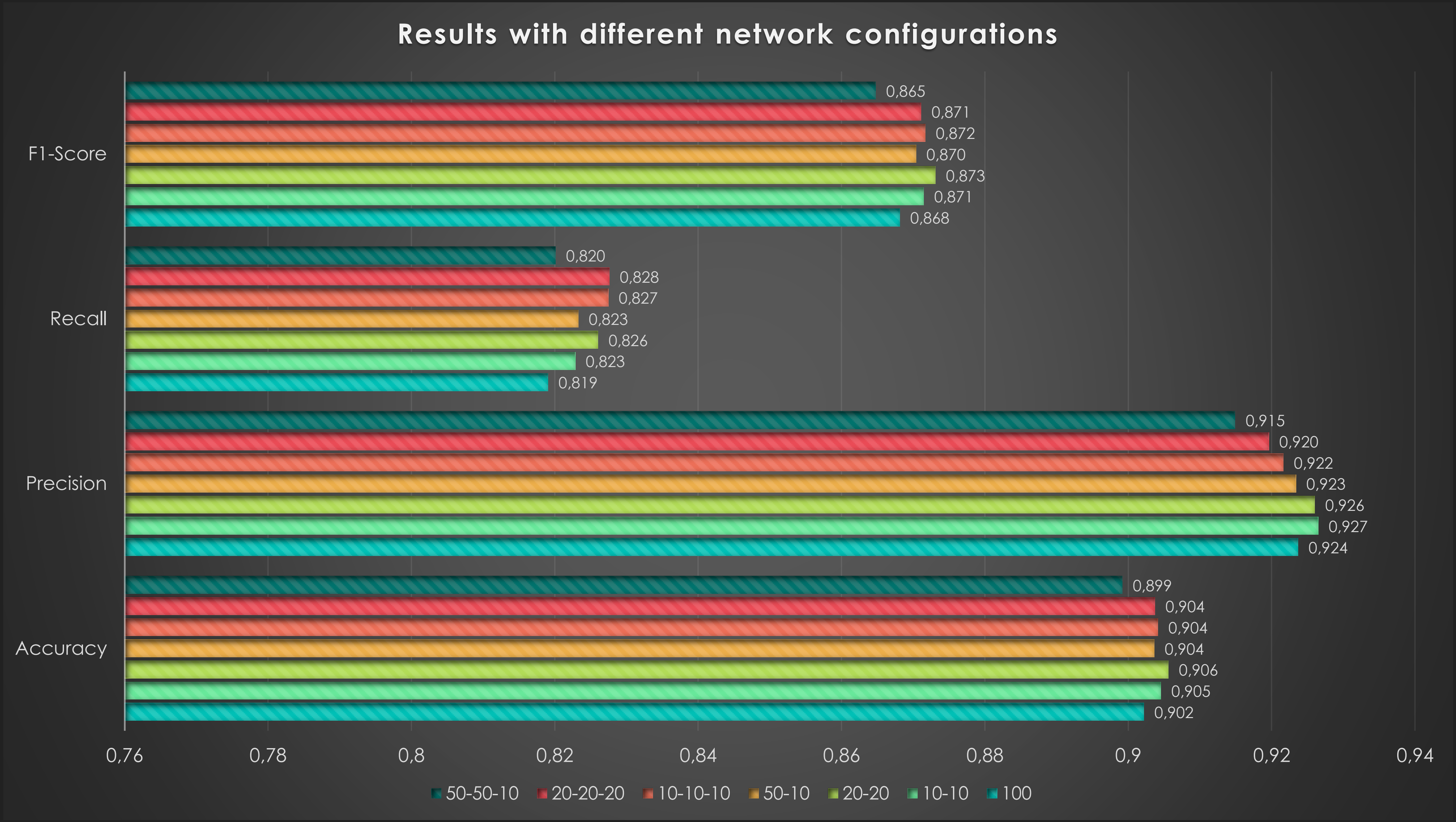

Average accuracy with different number of most used tokens
AVERAGE ACCURACY
0,9
0,85
0,8
0,75
0,7
0,65
0,6
0
5000
10000
15000
20000
25000
NUMBER OF MOST USED TOKENS
1-gram-count
2-gram-count
1-2-gram-count
Tf-idf
1-gram-count_tfidf
2-gram-count_tf-idf
1-2-gram-count_tf-idf

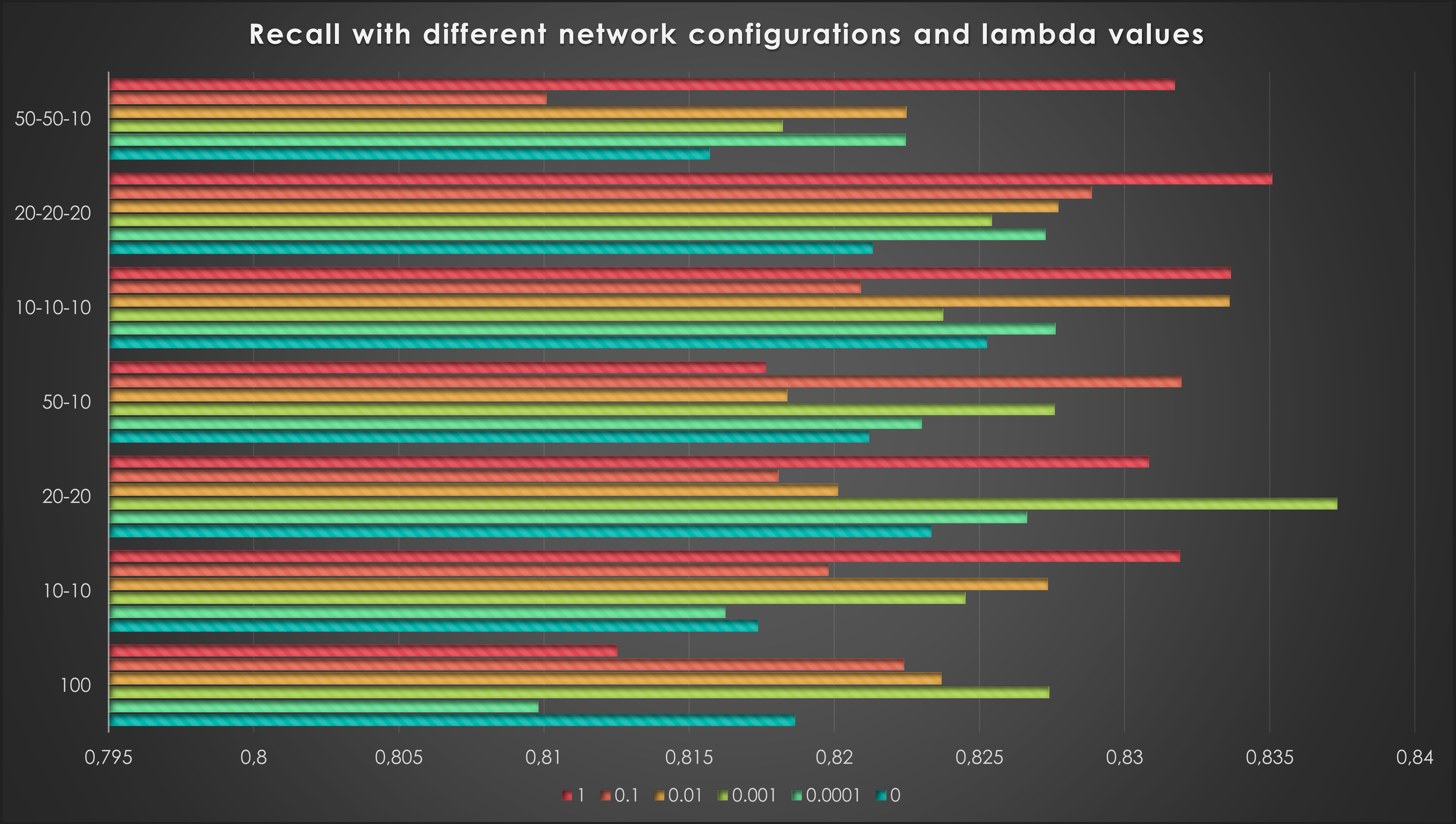
Results with 2-gram count and tf-idf features



Multi-layer Perceptrons (MLP)

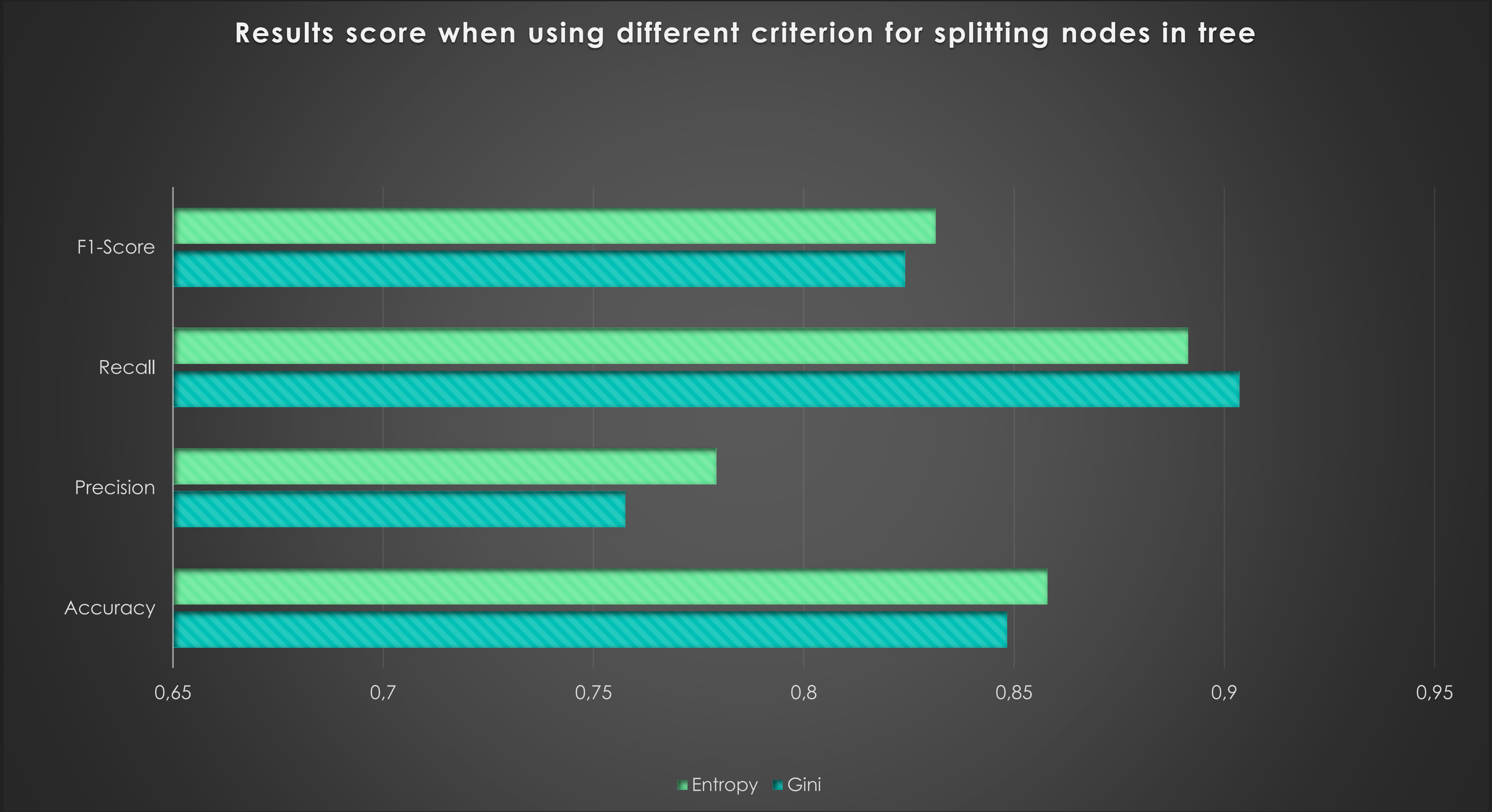
- Activation function: ReLU
- Solver: Adam
- 500 epochs
- Early Stopping activated
- Different network configurations tested:
  - 100, 10-10, 20-20, 50-10, 10-10-10, 20-20-20, 50-50-10
- Different lambda values tested:
  - 0, 0.0001, 0.001, 0.01, 0.1, 1

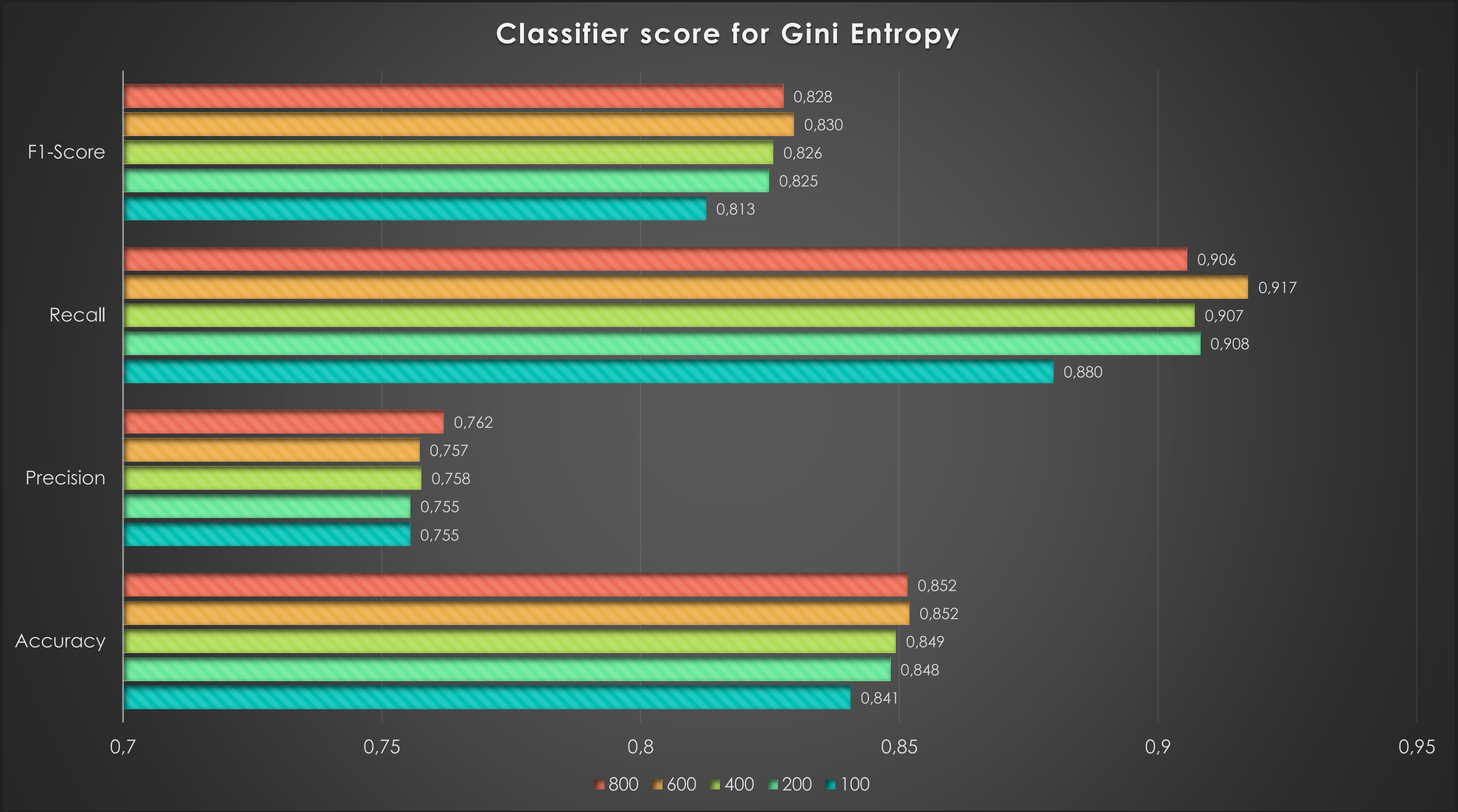

Results with different network configurations
F1-Score
0,865
0,871
0,872
0,870
0,873
0,871
0,868
Recall
0,820
0,828
0,827
0,823
0,826
0,823
0,819
Precision
0,915
0,920
0,922
0,923
0,926
0,927
0,924
Accuracy
0,899
0,904
0,904
0,904
0,906
0,905
0,902
0,76
0,78
0,8
0,82
0,84
0,86
0,88
0,9
0,92
0,94
50-50-10
20-20-20
10-10-10
50-10
20-20
10-10
100


Recall with different network configurations and lambda values
50-50-10
20-20-20
10-10-10
50-10
20-20
10-10
100
0,795
0,8
0,805
0,81
0,815
0,82
0,825
0,83
0,835
0,84
1
0.1
0.01
0.001
0.0001
0

Decision Tree

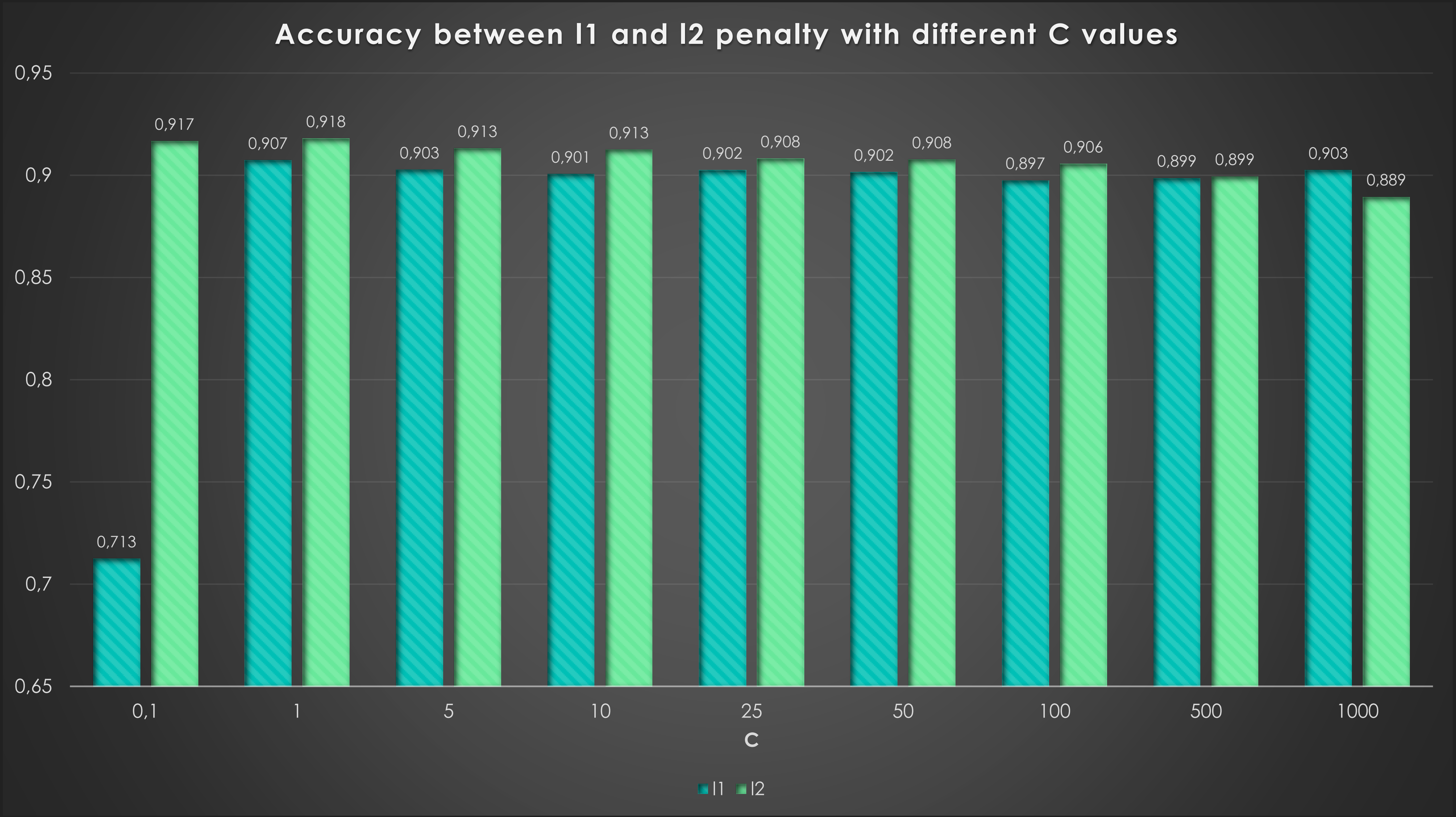
- Criterion: measures the quality of a split
  - Gini
  - Entropy
- Maximum depth of decision tree
  - 100
  - 200
  - 400
  - 600
  - No maximum depth (~800)


Results score when using different criterion for splitting nodes in tree
F1-Score
Recall
Precision
Accuracy
0,65
0,7
0,75
0,8
0,85
0,9
0,95
Entropy
Gini

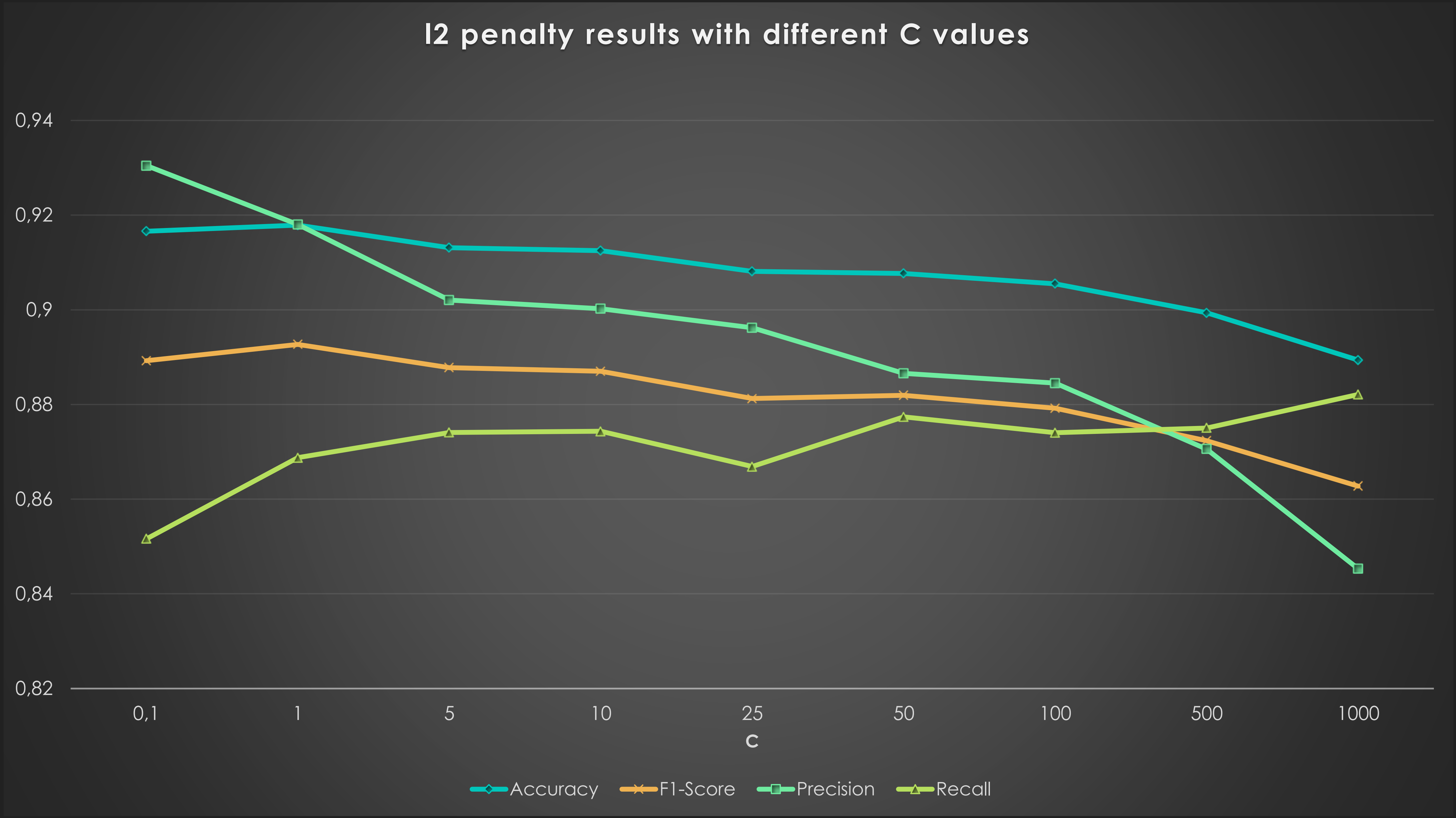

Classifier score for Gini Entropy
F1-Score
0,828
0,830
0,826
0,825
0,813
Recall
0,906
0,917
0,907
0,908
0,880
Precision
0,762
0,757
0,758
0,755
0,755
Accuracy
0,852
0,852
0,849
0,848
0,841
0,7
0,75
0,8
0,85
0,9
0,95
800
600
400
200
100

Linear SVC

- Penalty: norm used in the penalization
  - l1
  - l2
- C: penalty parameter C of the error term
  - 0.1
  - 1
  - 5
  - …
  - 1000

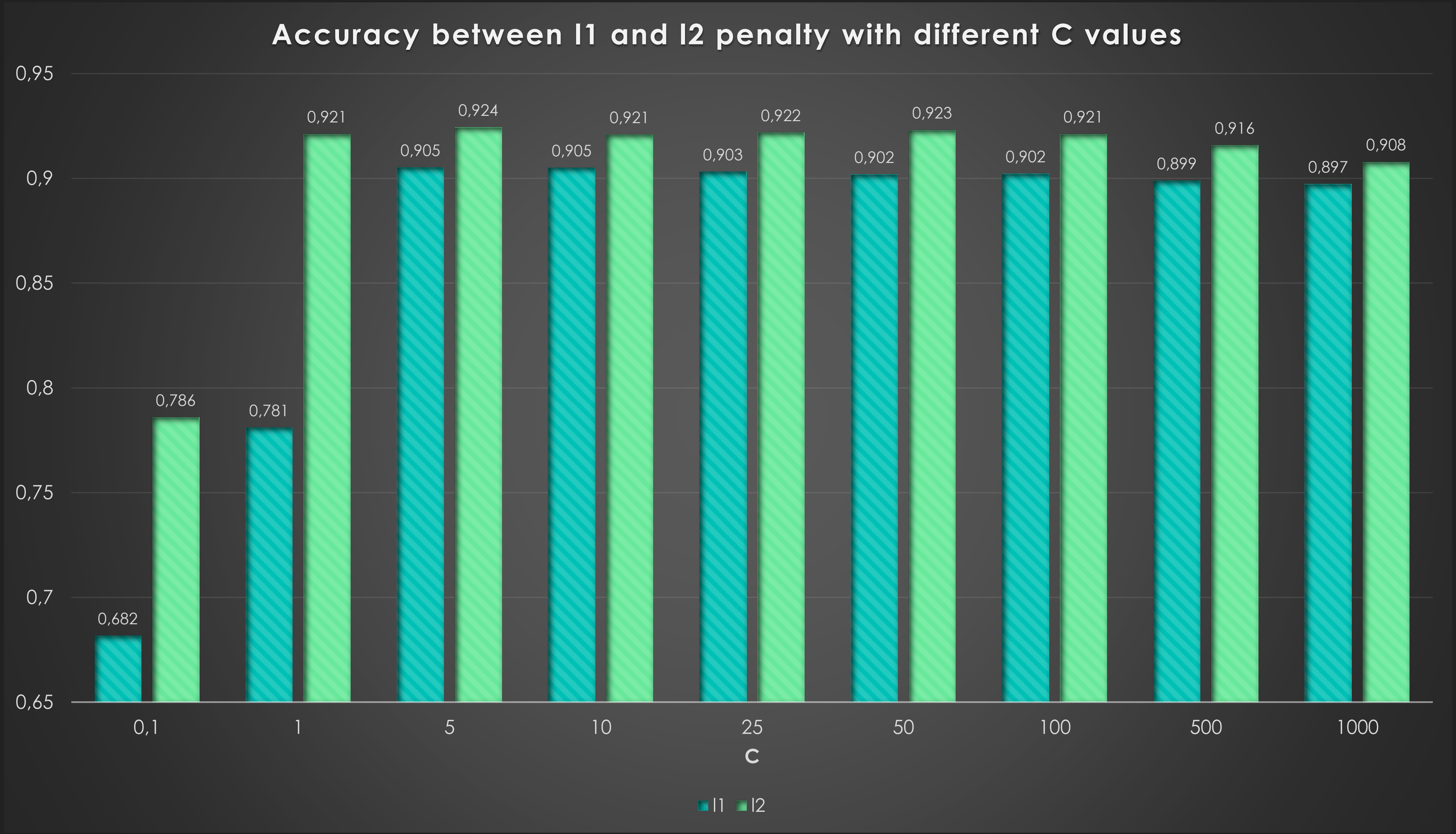


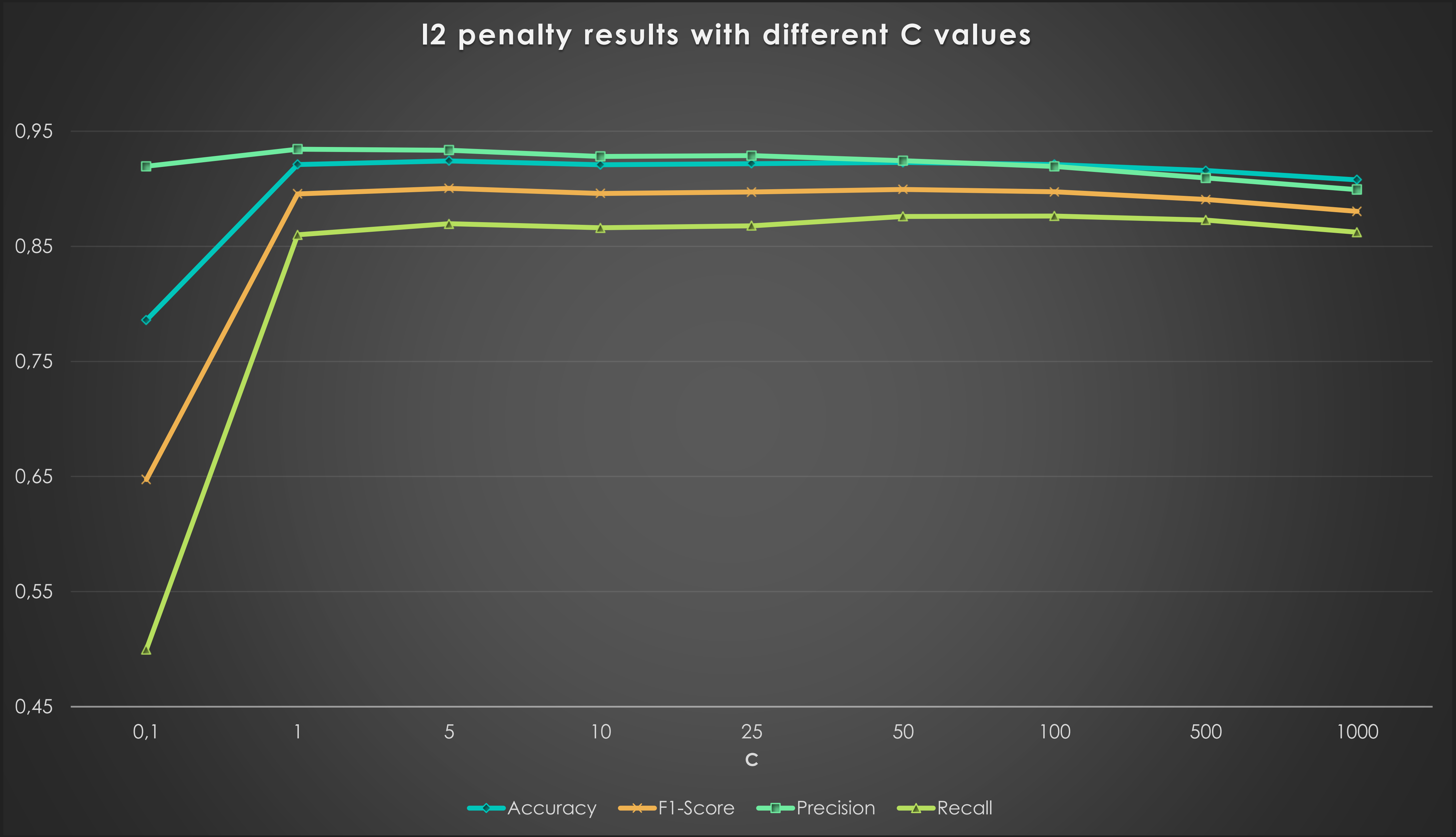
Accuracy between l1 and l2 penalty with different C values
0,95
0,9
0,85
0,8
0,75
0,7
0,65
0,713
0,917
0,907
0,918
0,903
0,913
0,901
0,913
0,902
0,908
0,902
0,908
0,897
0,906
0,899
0,899
0,903
0,889
0,1
1
5
10
25
50
100
500
1000
C
l1
l2


l2 penalty results with different C values
0,94
0,92
0,9
0,88
0,86
0,84
0,82
0,1
1
5
10
25
50
100
500
1000
C
Accuracy
F1-Score
Precision
Recall

Logistic Regression

- Penalty: norm used in the penalization
  - l1
  - l2
- C: penalty parameter C of the error term
  - 0.1
  - 1
  - 5
  - …
  - 1000


Accuracy between l1 and l2 penalty with different C values
0,95
0,9
0,85
0,8
0,75
0,7
0,65
0,682
0,786
0,781
0,921
0,905
0,924
0,905
0,921
0,903
0,922
0,902
0,923
0,902
0,921
0,899
0,916
0,897
0,908
0,1
1
5
10
25
50
100
500
1000
C
l1
l2


l2 penalty results with different C values
0,95
0,85
0,75
0,65
0,55
0,45
0,1
1
5
10
25
50
100
500
1000
C
Accuracy
F1-Score
Precision
Recall

Final results

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.956	0.940	0.946	0.943
Linear SVC	0.952	0.924	0.952	0.938
Decision Tree	0.871	0.764	0.959	0.850
Multi-layer Perceptrons	0.944	0.943	0.907	0.925

Data division: 80% train and 20% test