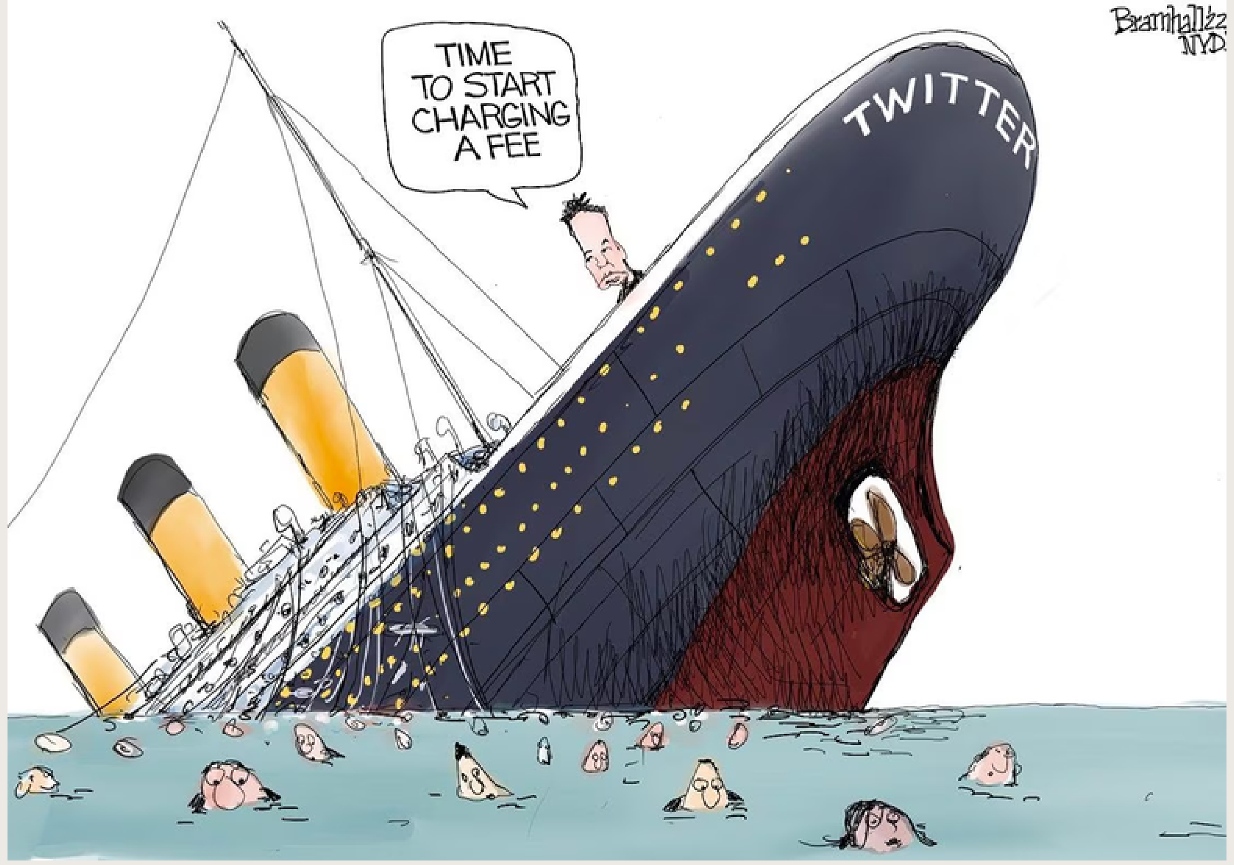
DWM PROJECT

PROBLEM DESCRIPTION

To create a model that predicts which passengers survived the Titanic shipwreck.

“what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc).
why this dataset and applications of it:-



1. Rich dataset: The Titanic dataset is a rich dataset that contains various types of data such as categorical, numerical, and textual data. This allows for a wide range of analysis and modeling techniques to be applied to the dataset.
2. Availability of data: The Titanic dataset is readily available and easily accessible online. This makes it an ideal dataset for beginners to practice and learn data mining and machine learning techniques.

BY RAHUL THAMBI, A059

PROBLEM DESCRIPTION

3. **Real-world application:** The Titanic dataset is based on a real-world event and has practical applications. For example, analyzing the data can help identify factors that contributed to the survival of passengers and provide insights for improving safety measures in future maritime disasters.
4. **Benchmark dataset:** The Titanic dataset is often used as a benchmark dataset for evaluating and comparing different data mining and machine learning algorithms. This is because it is a well-known and well-studied dataset, and there are many existing results that can be used for comparison.
5. **Visualization opportunities:** The Titanic dataset is a relatively small dataset, which makes it easy to visualize and explore. This provides opportunities for data analysts to create informative and engaging visualizations that can help communicate insights to stakeholders.

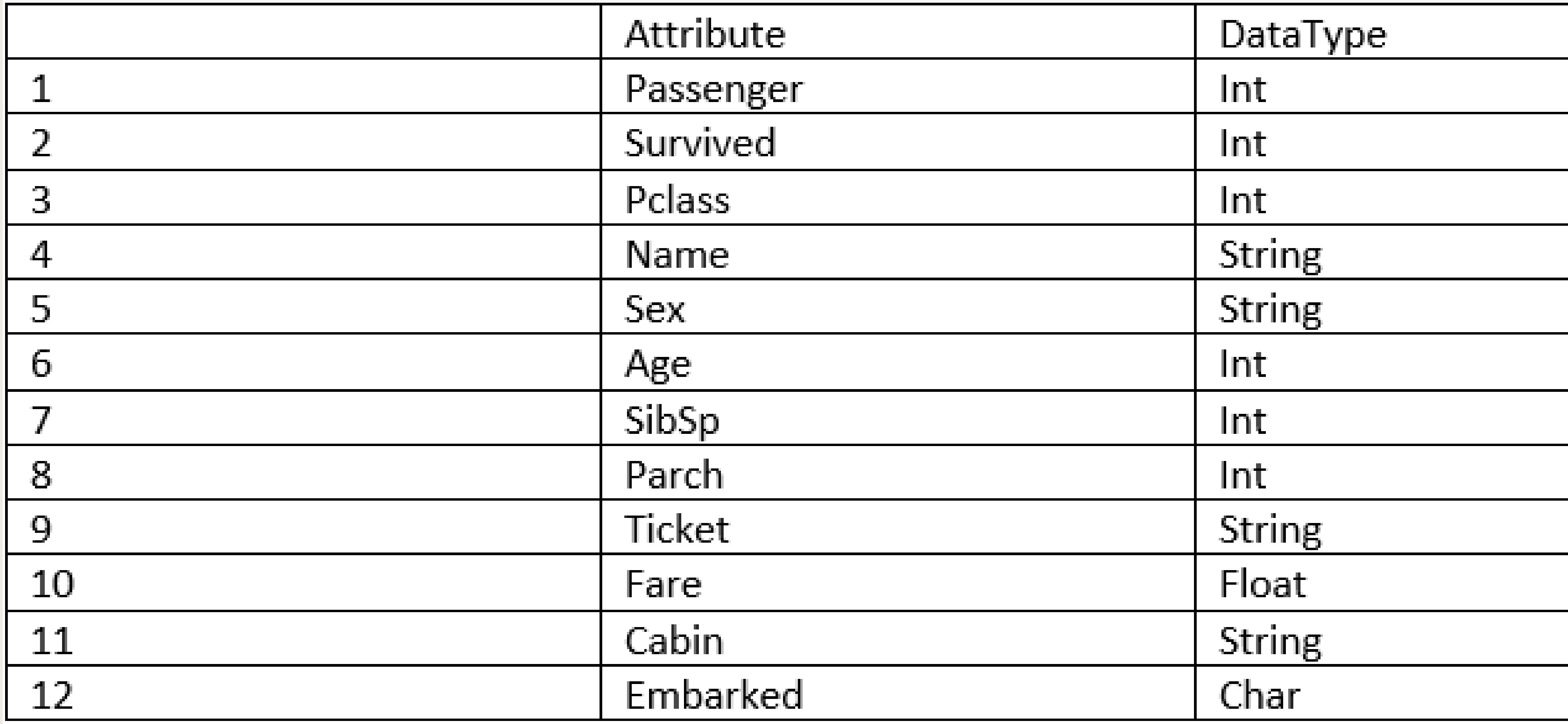
DATASET DESCRIPTION

DATASET NAME: TITANIC

12 ATTRIBUTES and 891 rows in total, Wherein "Survived" is the class attribute.

1. PassengerId: A unique identifier assigned to each passenger.
2. Survived: Whether or not the passenger survived the sinking of the Titanic (0 = No, 1 = Yes).
3. Pclass: The passenger's ticket class (1 = 1st, 2 = 2nd, 3 = 3rd).
4. Name: The passenger's name.
5. Sex: The passenger's sex (male or female).
6. Age: The passenger's age in years.
7. SibSp: The number of siblings/spouses the passenger had aboard the Titanic.
8. Parch: The number of parents/children the passenger had aboard the Titanic.
9. Ticket: The passenger's ticket number.
10. Fare: The fare paid by the passenger.
11. Cabin: The passenger's cabin number.
12. Embarked: The port where the passenger embarked (C = Cherbourg, Q = Queenstown, S = Southampton).

link to the dataset: https://www.kaggle.com/c/titanic



	Attribute	DataType
1	Passenger	Int
2	Survived	Int
3	Pclass	Int
4	Name	String
5	Sex	String
6	Age	Int
7	SibSp	Int
8	Parch	Int
9	Ticket	String
10	Fare	Float
11	Cabin	String
12	Embarked	Char

ALGORITHMS IDENTIFIED AND IMPLEMENTED

Classification Algorithms:

NOTE:- For all the algorithms 10-Folds Cross-validation is used.

The below algorithm performed the highest accuracy among all other algorithms selected (7).

The other algorithms were OneR :- 67.9415%, RandomTree:- 67.7165%, RandomForest:- 69.0664%

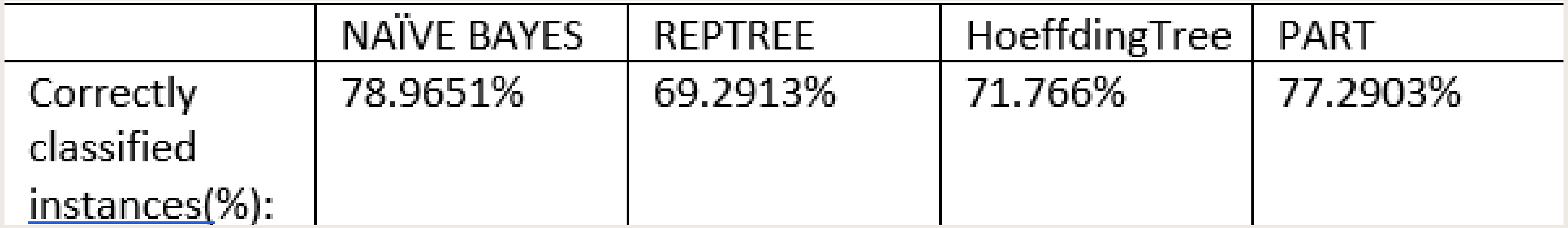
Algorithm chosen finally:-

1) **NAÏVE BAYES ALGORITHM:**

1) **REPTREE:**

3) **PART**

4) **HoeffdingTree**



	NAÏVE BAYES	REPTREE	HoeffdingTree	PART
Correctly classified instances(%):	78.9651%	69.2913%	71.766%	77.2903%

link to the algorithm descriptions:

https://scikit-learn.org/stable/index.html.

https://www.cs.waikato.ac.nz/ml/weka/.

https://www.kdnuggets.com/.

https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/PART.html

STEPS TAKEN TO IMPROVE THE ACCURACY

Since NaiveBayes gave the highest accuracy among all the other algorithms. Hence, I decided to increase its accuracy even further:-
before increasing the accuracy it's 78.9651%

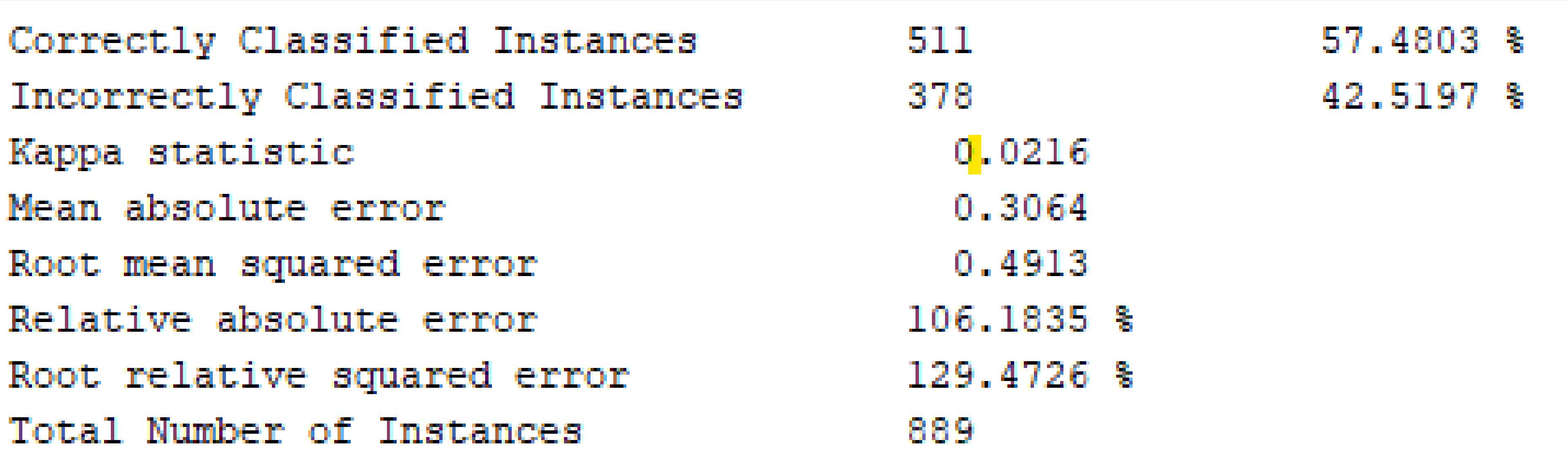
Step 1: preprocessing the data (removing redundant attributes)
-> "Name" attribute was removed since it didn't contribute to the class attributes value.

Step 2: Data Cleaning (missing values)
-> In total 37 rows had missing values, All the rows had "Age" as the missing value, so i calculated the mean value of the Age attribute and replaced that with the missing values.

STEPS TAKEN TO IMPROVE THE ACCURACY

Step 3: Converting the class attributes value from integer to Nominal
-> Where class attribute ("Survived"), had numerical values (1/0) which I converted into nominal values of (Yes/No), respectively. Since, NaiveBayes works best for nominal values rather than numerical.

Step 4: There were many ensemble technique I tried to increase the accuracy of NaiveBayes
1) adaBoostM1 (This technique rather decreased the accuracy from 78.96% to 57.48%)

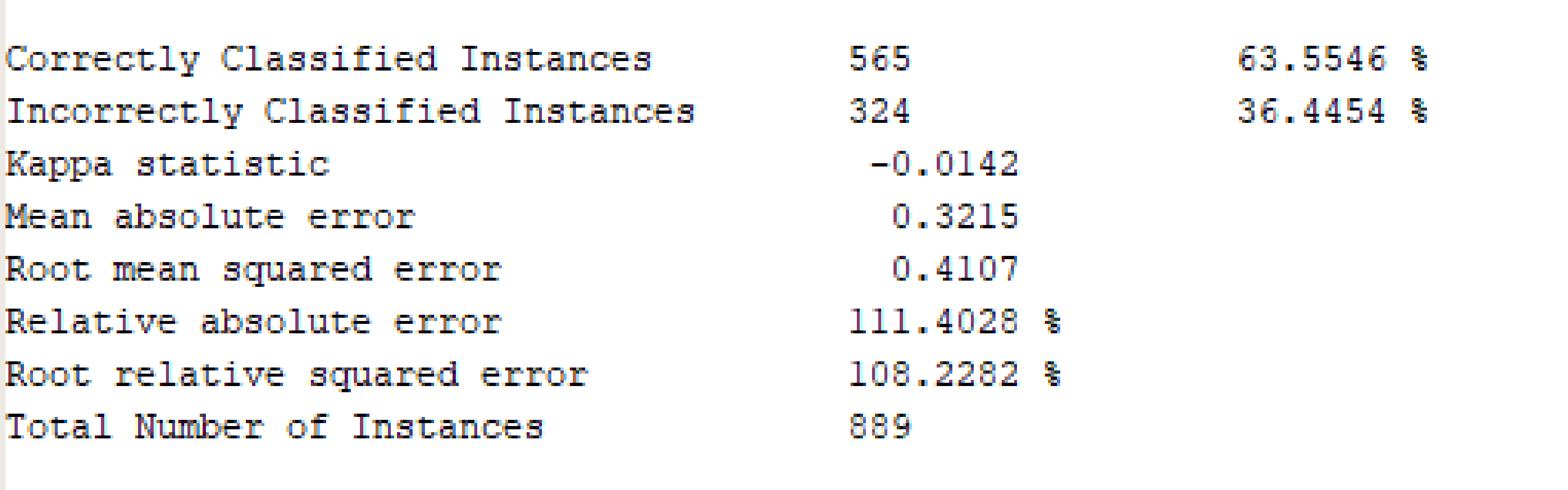


```
Correctly Classified Instances         511               57.4803 %
Incorrectly Classified Instances       378               42.5197 %
Kappa statistic                          0.0216
Mean absolute error                      0.3064
Root mean squared error                  0.4913
Relative absolute error                106.1835 %
Root relative squared error            129.4726 %
Total Number of Instances              889
```

Therefore, Rejected.

STEPS TAKEN TO IMPROVE THE ACCURACY

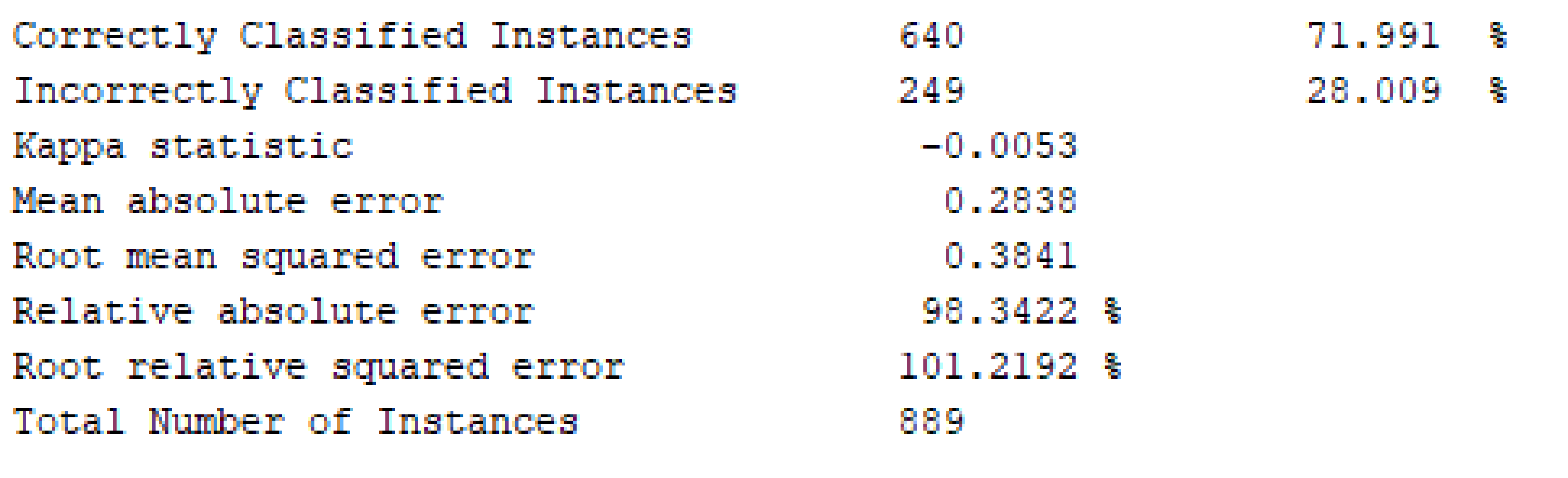
2) Bagging (This technique also decreased the accuracy from 78.96% to 63.55%)



```
Correctly Classified Instances         565               63.5546 %
Incorrectly Classified Instances       324               36.4454 %
Kappa statistic                         -0.0142
Mean absolute error                      0.3215
Root mean squared error                  0.4107
Relative absolute error                111.4028 %
Root relative squared error            108.2282 %
Total Number of Instances              889
```

Therefore, Rejected.

3) Boosting (This technique also decreased the accuracy from 78.96% to 71.99%)

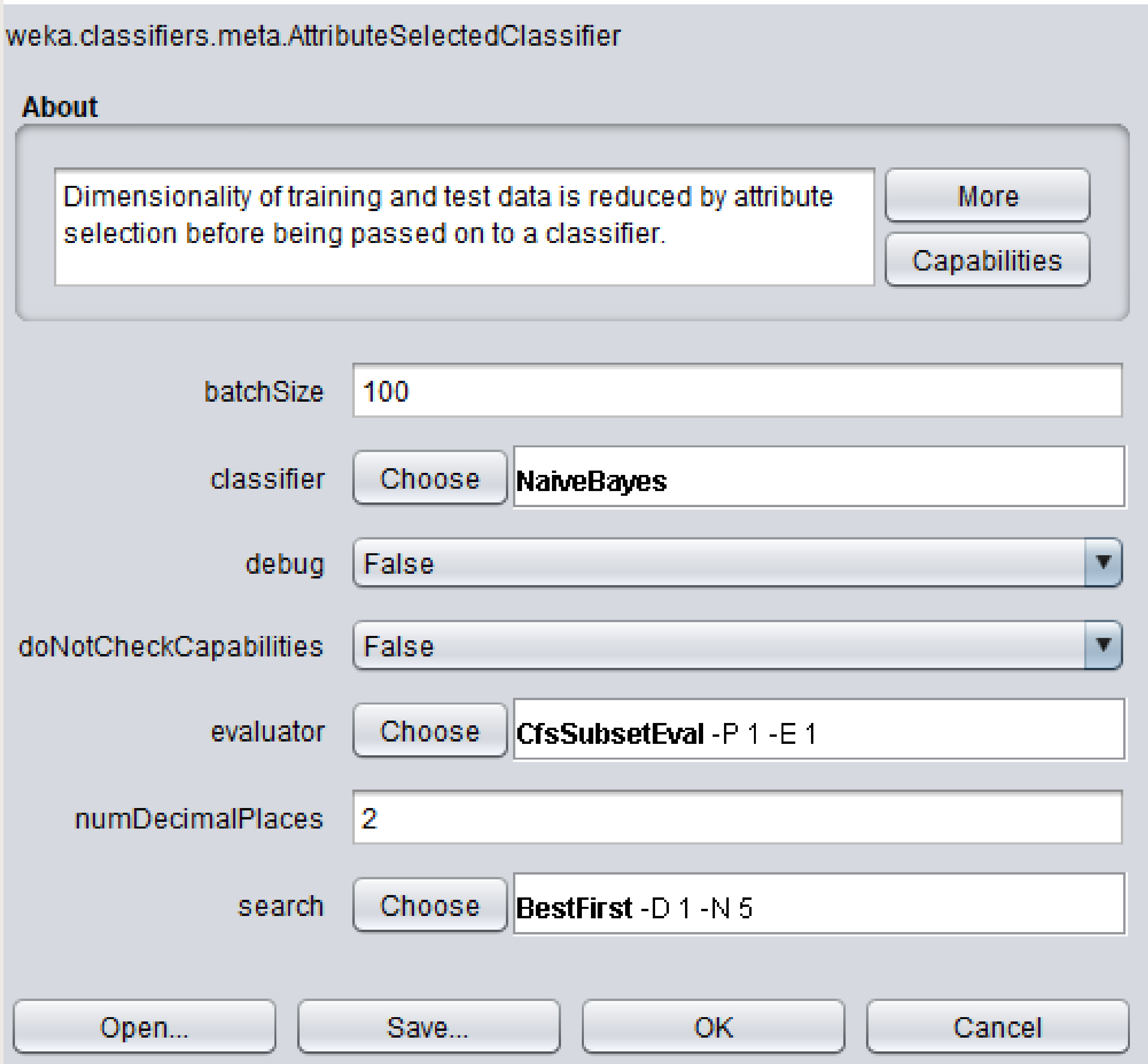


```
Correctly Classified Instances         640               71.991  %
Incorrectly Classified Instances       249               28.009  %
Kappa statistic                         -0.0053
Mean absolute error                      0.2838
Root mean squared error                  0.3841
Relative absolute error                 98.3422 %
Root relative squared error            101.2192 %
Total Number of Instances              889
```

Therefore, Rejected.

STEPS TAKEN TO IMPROVE THE ACCURACY

4) Finally, AbsoluteSelectedClassifier helped in increasing the accuracy from 78.96% to 81.21% i.e. an increment of 2.25%


weka.classifiers.meta.AttributeSelectedClassifier

About

Dimensionality of training and test data is reduced by attribute selection before being passed on to a classifier.

batchSize: 100

classifier: NaiveBayes

debug: False

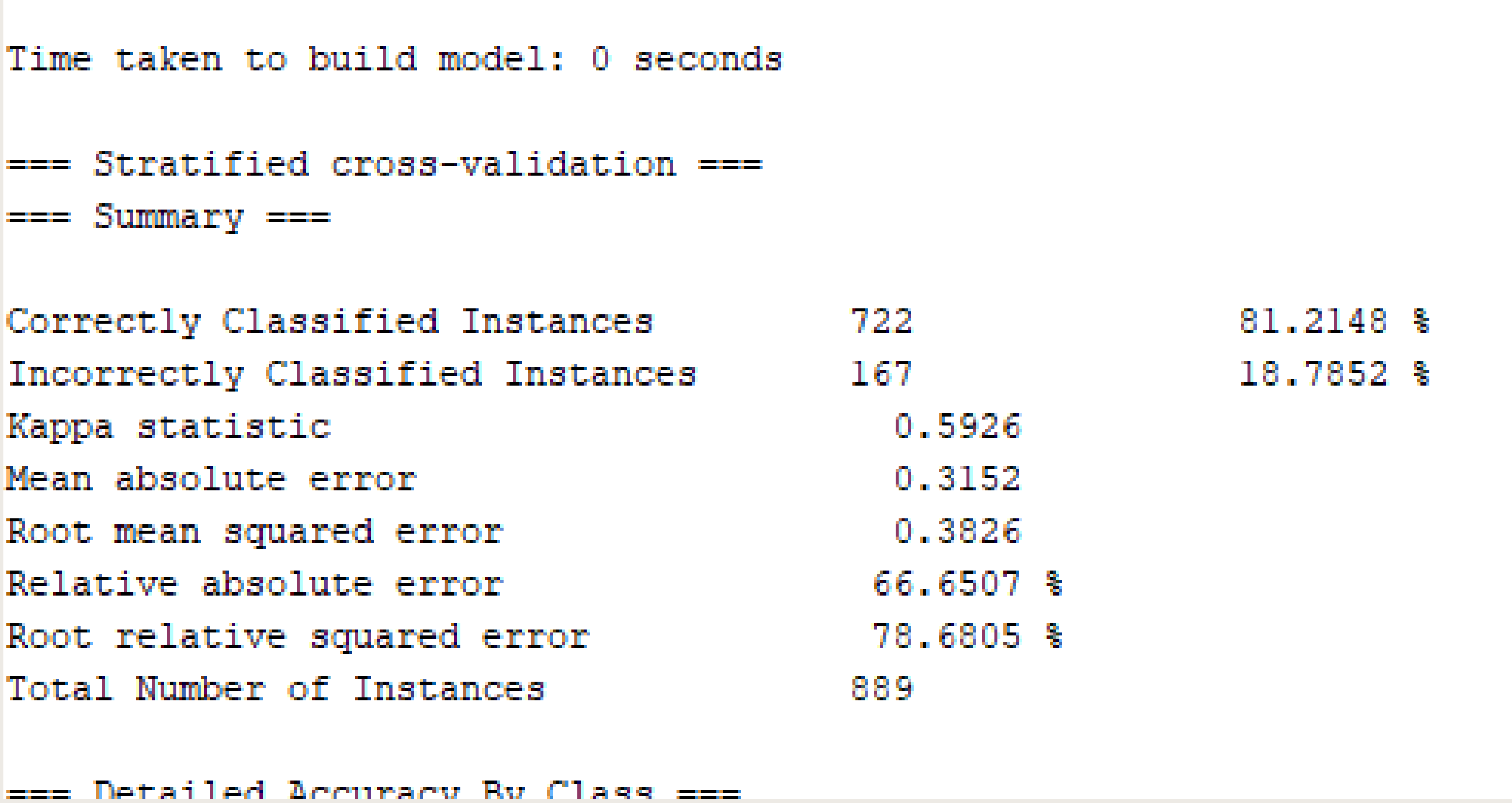
doNotCheckCapabilities: False

evaluator: CfsSubsetEval -P 1 -E 1

numDecimalPlaces: 2

search: BestFirst -D 1 -N 5

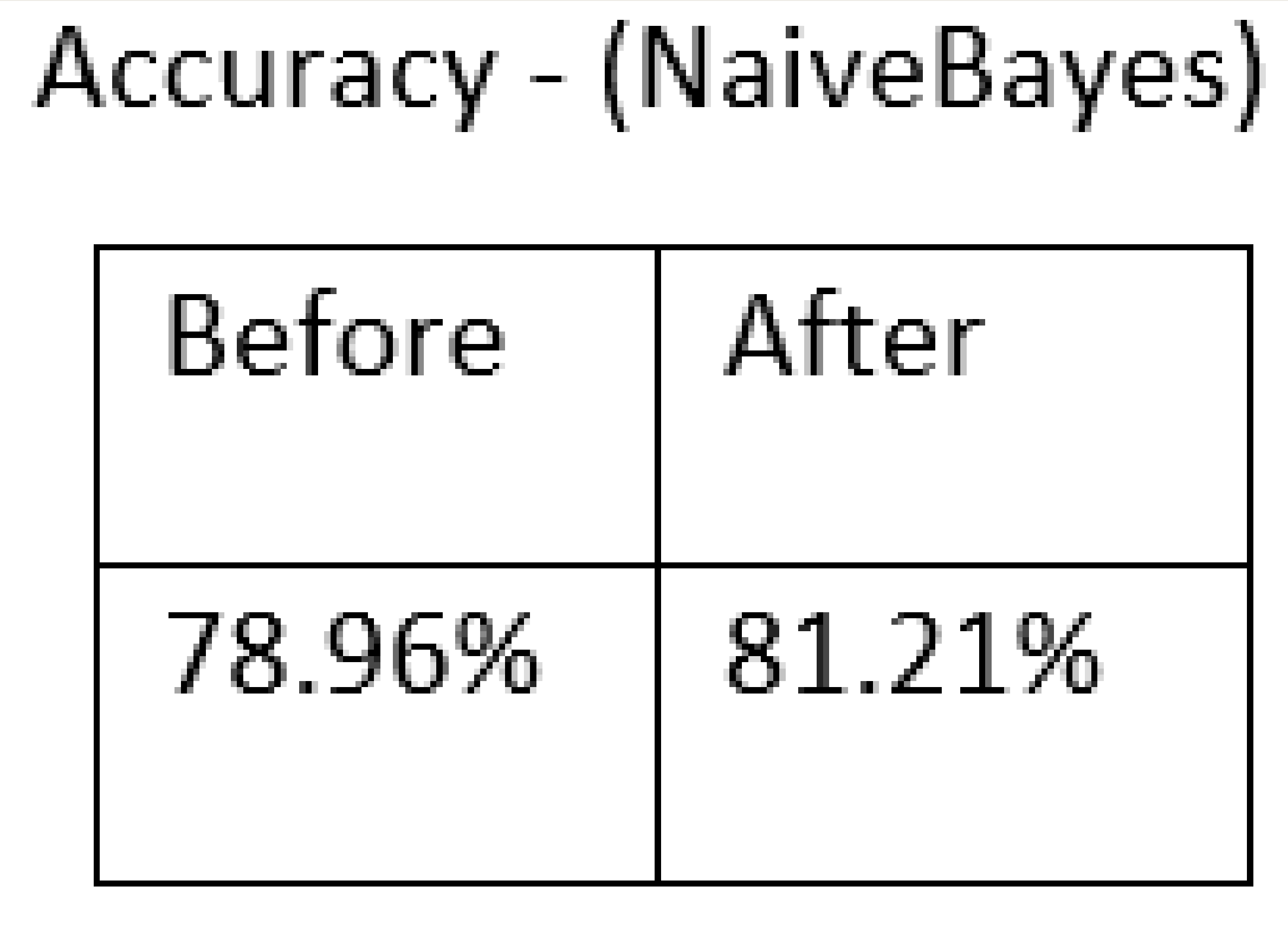
Where the classifier chosen was "NaiveBayes"
the evaluator was "cfsSubsetEval"
and search techinique was "BestFirst"


```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         722               81.2148 %
Incorrectly Classified Instances       167               18.7852 %
Kappa statistic                          0.5926
Mean absolute error                      0.3152
Root mean squared error                  0.3826
Relative absolute error                 66.6507 %
Root relative squared error             78.6805 %
Total Number of Instances              889

=== Detailed Accuracy By Class ===
```

STEPS TAKEN TO IMPROVE THE ACCURACY

Where AbsoluteSelectedClassifier is a feature selection technique that ranks the features in a dataset based on their absolute correlation with the target variable. It selects a fixed number of top-ranked features and discards the remaining features. By selecting the most important features, AbsoluteSelectedClassifier can improve the performance of a machine learning model by reducing the noise and focusing on the most informative features.

Wherein, the evaluator is used to evaluate the performance of different subsets of features (here, I have chosen CfsSubsetEval, since it gave a better performace than others) and search is and algorithm that is used to search through the space of possible feature subset of features (here, I have chosen BestFirst, since it gave a better performace than others)



Accuracy - (NaiveBayes)

Before	After
78.96%	81.21%

link referred:
https://www.kaggle.com/code/vinothan/titanic-model-with-90-accuracy

REFERENES:-

- https://www.kaggle.com/c/titanic
- https://scikit-learn.org/stable/index.html.
- https://www.cs.waikato.ac.nz/ml/weka/.
- https://www.kdnuggets.com/.
- https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/PART.html
- https://www.kaggle.com/code/vinothan/titanic-model-with-90-accuracy

ACKNOWLEDGEMENT:-

- Special thanks to https://github.com/vinothhunt, for providing the various steps to increase the accuracy of the algorithm used (NaiveBayes).
- Websites like kaggle, sourceforge had a major role in the making of this project possible.

THANKYOU