Assignment 2

Machine learning DV2578

Suresh Banduru

Telecommunication

Blekinge Tekniska Hogskola

Karlskrona, Blekinge suba21@student.bth.se

# **I. INTRODUCTION**

The assignment is based on experimenting the three learning algorithm and comparing their performance. The evolution measure is based on the three aspects. First compare it with the computational performance on the basis of training time and second predictive performance on the basis of accuracy and the third predictive performance on the basis of F-measure.

# **II. TASK 1**

## **Data Preprocessing Model (Spambase dataset)**

**Dataset Summary**

Number of rows: 4600

Number of columns: 58

Problem type: Binary Classification

Duplicate rows: 8.50%

High! Severity Insight: Duplicate Rows Detected

Severity: High

**Issue:**

Approximately 8.50% of the rows in the dataset are found to be duplicates. Duplicate rows can significantly impact the accuracy and reliability of data analysis, leading to skewed results and erroneous conclusions.

**Action Item**

Identify and remove duplicate rows from the dataset. This process will improve the quality of the dataset and enhance the effectiveness of any downstream analysis or modeling tasks.

Missing target values 0.00%

Invalid target values 0.04%

**Valid values**

**Numeric features**

All values that could be cast to finite floats are valid. Missing values are not valid.

Column Type Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Categories** | **Type** | **Count** | **Percentage** |
| **0** | **Numeric** | **float64** | **55** | **94.827586** |
| **1** | **Numeric** | **int64** | **3** | **5.172414** |

**Target Analysis**

The column spamorham is used as the target column. See the distribution of target column values (labels) in the target column below:

Number of Classes: 2

Adeela Abid

MS in Computer Science

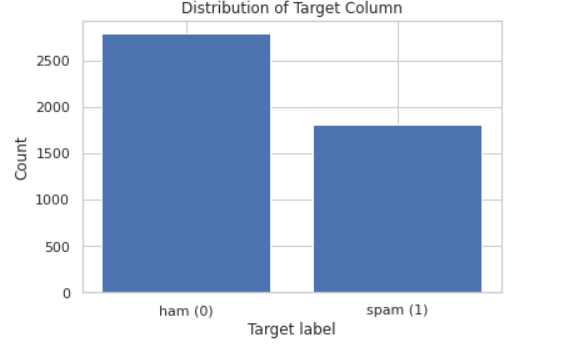
Blekinge Tekniska Hogskola

Karlskrona, Blekinge

adab22@student.bth.se

Positive Label: spam (1)

Negative Label: ham (0)



**Missing Values**

Empty strings and strings composed of only white spaces are considered missing.

Missing target values: 0.00%

**Invalid values**

Values that are either missing or that could not be cast to the desired type.

Invalid target values: 0.00%

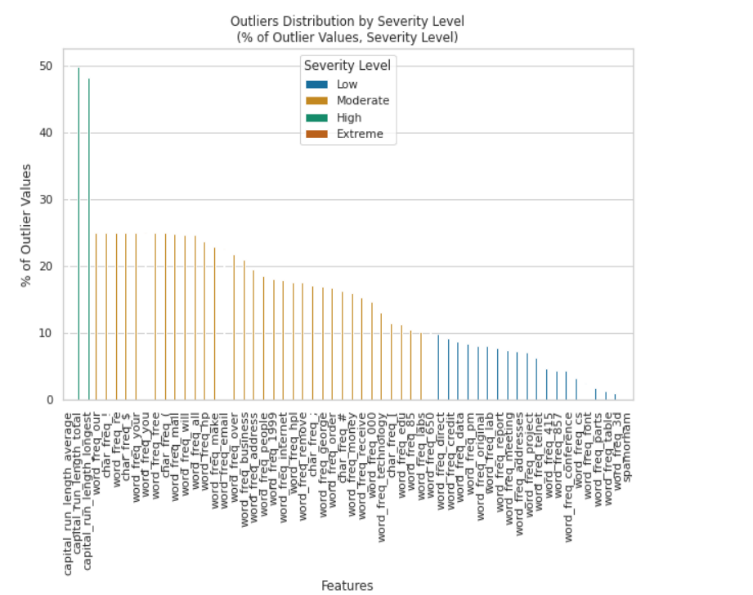
**Descriptive Statistics**

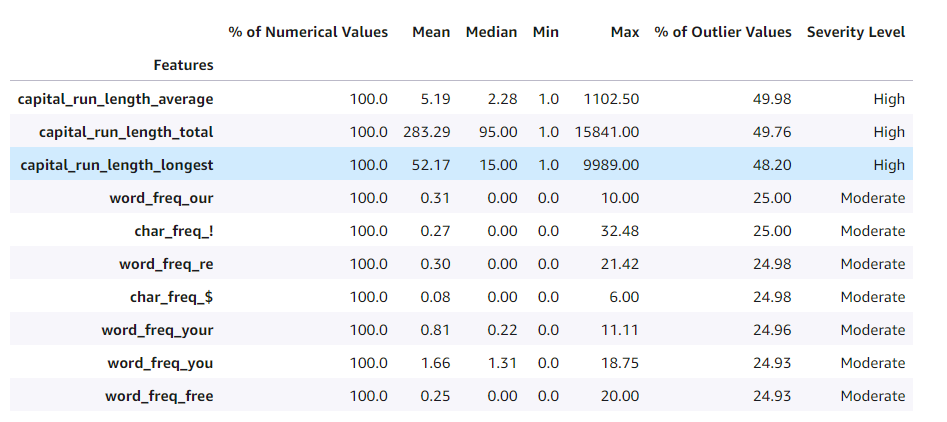
The Descriptive statistics are computed from the data sample.

We found 58 of the 58 columns contained at least one numerical value.

**Outliers**

The outlined strategy provides a systematic approach for identifying and analyzing outliers in a dataset. The steps involve calculating the percentage of outliers by comparing values to percentiles, selecting the top outliers, creating a comprehensive table with statistics, sorting it by outlier percentages, adding a severity level column, and resetting the index for clarity. This approach quantifies and categorizes outliers, facilitating further analysis and decision-making based on their severity across different features.





**Action Items:**

* Investigate the origin of the data field.
* Are some values non-finite (e.g., infinity, nan)?
* Are they missing or is it an error in data input?
* Missing and extreme values may indicate a bug in the data collection process.
* Verify the numerical descriptions align with expectations.
* Use domain knowledge to check that the range of values for a feature meets expectations.

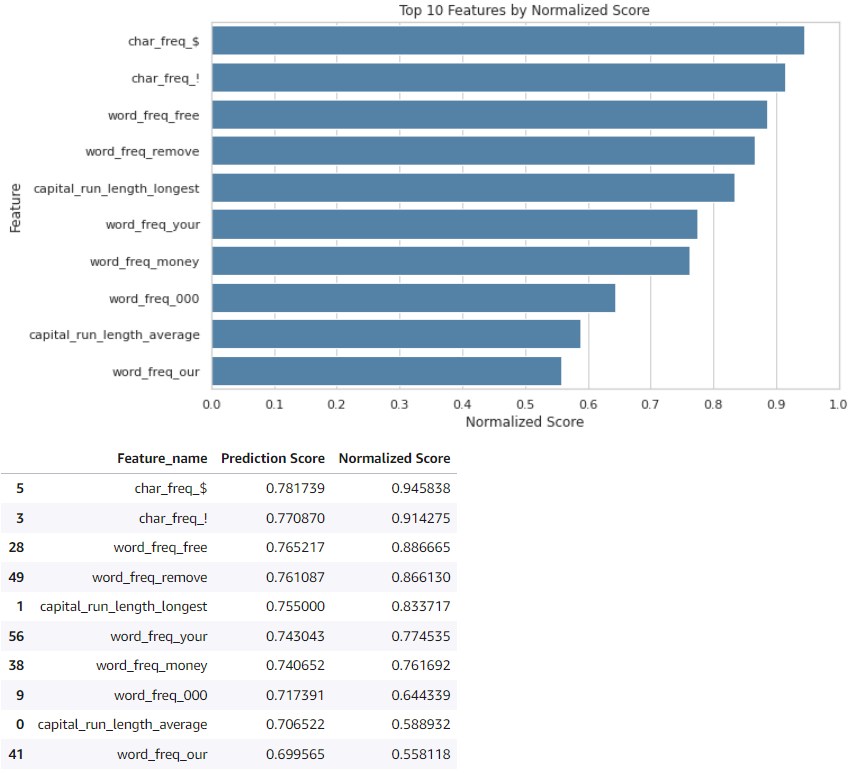
**Feature Summary**

Prediction power is measured by stratified splitting the spam data set into 80% and 20% training and validation folds.

We fit a Random Forest Classifier model for each feature separately on the training fold after applying minimal feature pre-processing and measure prediction performance on the validation data.

Higher prediction power scores, toward 1, indicate columns that are more useful for predicting the target on their own. Lower scores, toward 0, point to columns that contain little useful information for predicting the target on their own.

Note: We are only showing the top 10 predicted results.



## **Evaluation Strategy**

1. We performed the stratified ten-fold cross-validation test.
2. Randomly partition the data in k parts or 'folds' (k=10)
3. Set one-fold aside for testing, train the model on the remaining 10-folds and **evaluate** it on the test fold.
4. These are the following results after we evaluated the three supervised learning algorithms with 10-fold cross-validation as shown in 12.4.

**Algorithms:**

1. Naïve Bayes

Naïve Bayes algorithm is based on the Bayes theorem used to predict the probability of an instance. It assumes the independence between the features. It is suitable for large feature spaces.

1. Decision Tree

Decision Tree algorithm is based on tree-like model to take decisions by splitting the data base on different features. It is famous algorithm for classification and regression tasks.

1. Nearest Neighbor

Nearest Neighbor is based on the concept of proximity to classifies or predicts new data points. It follows the rule that similar instance belongs to the same class. It is also suitable for classification and regression tasks but can be expensive in terms of computation for large data sets.

1. **Evaluation of the algorithms**

The three-evaluation measures we used to compare the performance of our all-listed algorithms on the base of performance are training time, accuracy and F-measure.

**Evaluate the Accuracy:**

Model accuracy: We calculated the model accuracy by

acc = (pos\*tpr)+(neg\*tnr)

Formula 1: Model accuracy, Classification performance table Summary as per Flach, Table 2.3

Where,

acc = accuracy,

pos = number of positives,

tpr = true positive rate,

neg = number of negatives and

tnr = true negative rate.

**Evaluate the F-measure:**

The next we calculated the F-measure or the harmonic mean of precision and recall (background 10.1) (F-measure)

F-measure = 2/((1/prec)+(1/rec))

Where,

Prec: We calculated the precision or confidence by

TP/(TP+FP)

TP = True Positives and FP = False Positives

We calculated the TP, FP by confusion matrix (Y training data, Y prediction data, class labels= [Spam (1), not a Spam (0)])

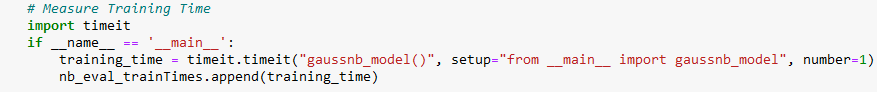
rec: We calculated the recall by TP/(TP+FN)

TP = True Positives and FN = False negatives

We calculated the TP, FN by confusion matrix (Y training data, Y prediction data, class labels = [Spam (1), not a Spam (0)])

**Evaluate the Execution Time:**

We measured the execution time of the model in seconds by python timeit module



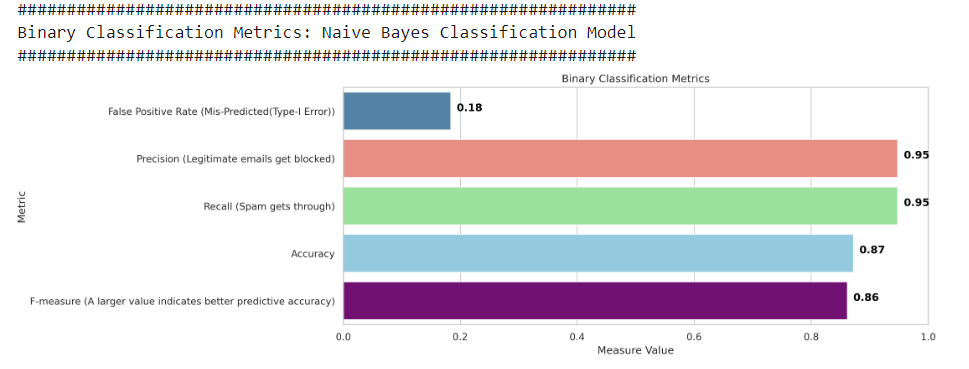
**The Accuracy Data and Rank Comparison of machine learning algorithms across multiple test folds:**

We use the sample accuracy data, assuming it comes from different data sets. The following table shows the ranks in brackets for three different machine learning algorithms (Naive Bayes, Decision Tree, Nearest Neighbour) across 10 folds:

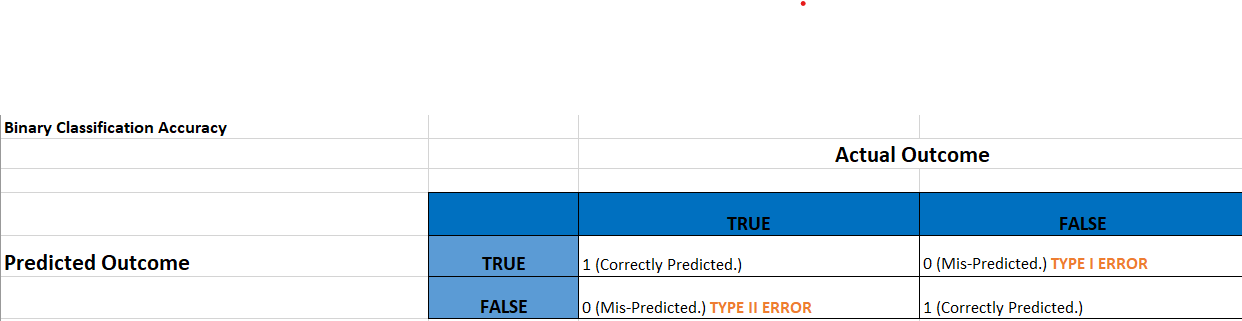
Algorithm Quality Report:

Naïve Bayes Algorithm

Metrics



The area under the receiver operating characteristic curve (AUC) is an industry standard accuracy metric used for binary classification problems. AUC measures the ability of the model to predict the best score for the positive labels (spam) as compared to negative labels (ham). The AUC metric returns a value from 0 to 1, mean AUC near 1 indicate the model that is highly accurate. The value near to 0 indicate a problem with the data.



**The correct prediction:**

True positive (TP), the model predicted the value as 1 and the true value is 1.

True negative (TN), the model predicted the value as 0 and the true value is 0.

Error predictions:

False positive (FP)

Type 2 error, our model predicted the value as 1 and true value is 0.

False Negative (FN)

Type 1 error, our model predicted the value as 0 and true value is 1

False positive rate:

The false positive rate (FPR) measures the fraction of actual negatives that are miss predicted as positives.

The false positive rate range is 0 to 1. The smaller value indicates the better predictive accuracy.

FPR = FP/(FP+TN)

True positive rate:

The true positive rate measures the fraction actual positives that are predicted as positive.

The TPR range is 0 to 1.

The larger value indicates better predictive accuracy.

TPR = TP/(TP+FM)

Precision recall curve:

The precision recall curve plots the trade of between the precision and recall.

The higher the precision and recall, the larger the area under the curve.

Precision:

Precision measures the fraction of actual positive that are predictive as positive. The precision range is 0 to 1.

The larger value indicates better accuracy in the value predicted.

Precision = TP/(TP+FP)

Recall:

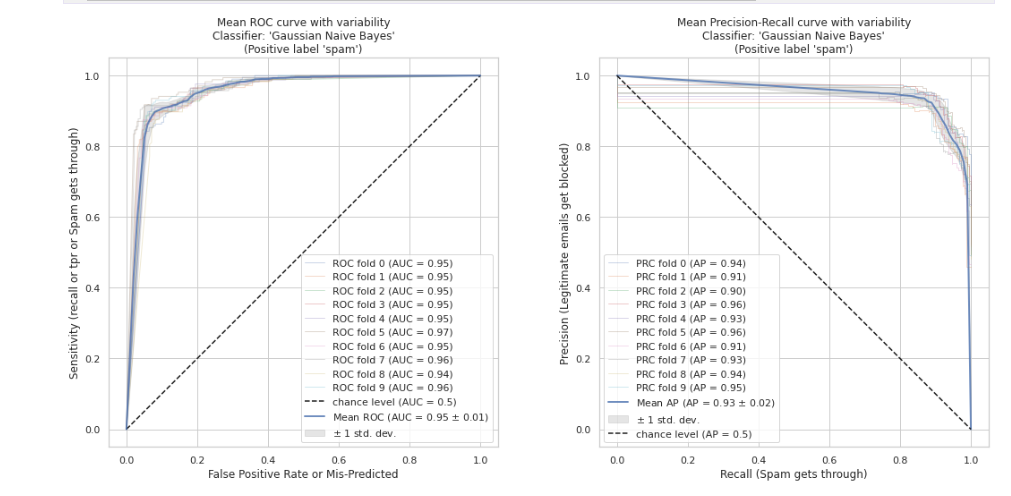
Recall measures the fraction of actual positive that are predicted as positive out of all the actual output in the test data set. This is also known as sensitivity or as the true positive rate.

The range is 0 to 1. The larger value indicates good detection of positive values from the test data set.

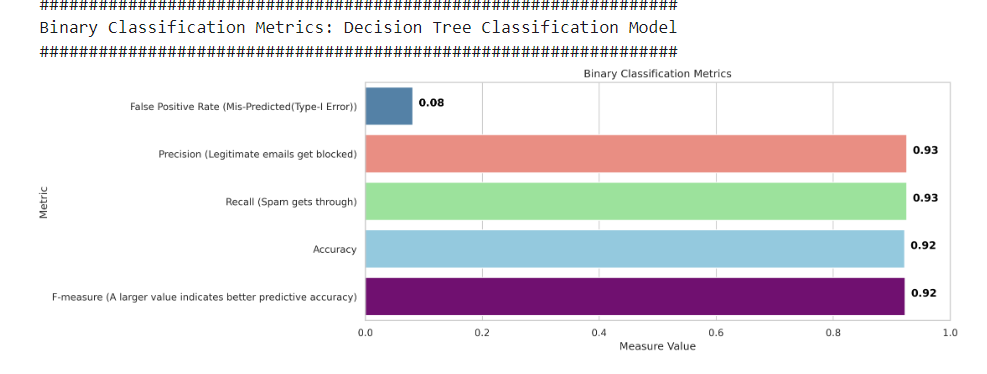
Recall = TP/(TP+FN)

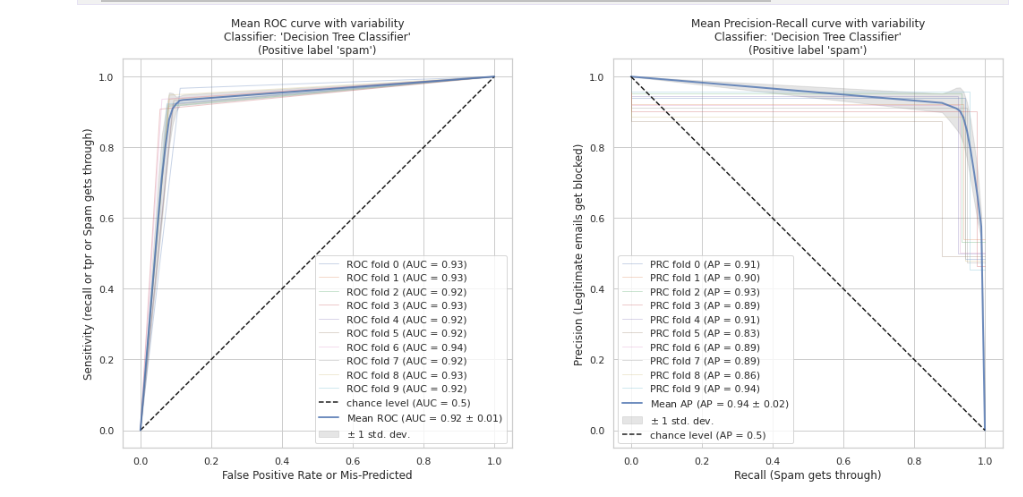
The precision recall curve has mean AP of 0.93 +/- 0.02.

The mean AUC of the model is 0.95 +/- 0.01.

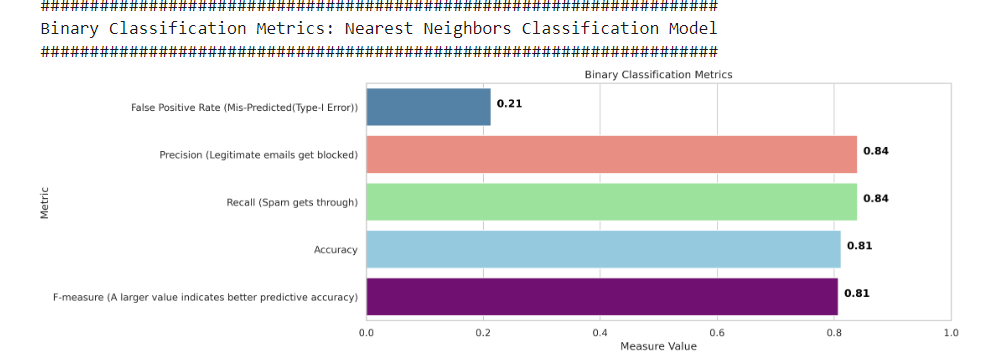


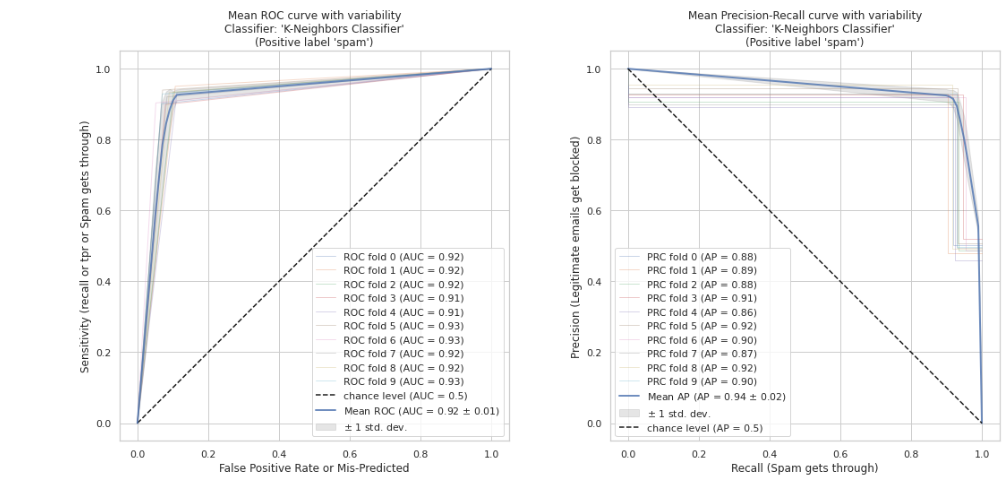
Decision Tree Algorithm

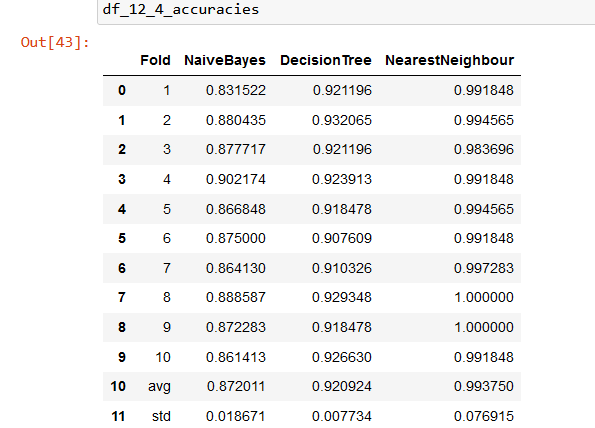


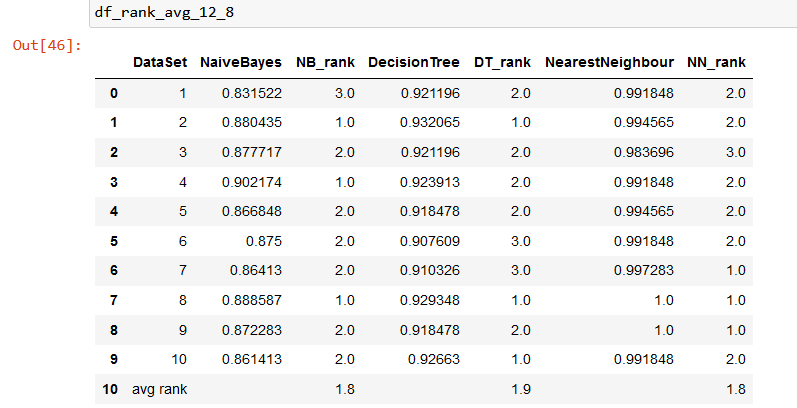


Nearest Neighbors Algorithm





  
Table 1: Accuracies; Cross-Validation Table (Example 12.4)

  
Table 2: Accuracies; Rank Table (Example 12.4)

1. **Friedman Test**

i) We define the Null and Alternative Hypotheses

The null and alternative hypotheses for the Friedman Test can be written as follows:

H0: All algorithms perform equally, and the average ranks are the same.

In mathematical form:

H0: ∀i,j(μi = μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that all algorithms have the same average rank.

H1: At least one of the algorithms performs differently than the others, and the average ranks are not all equal.

In mathematical form, the null and alternative hypotheses can be expressed as:

H1: ∃i,j(μi ≠ μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that at least one algorithm has a different average rank than the others.

Since our alternative hypothesis is H1: ∃i,j(μi ≠ μj), We consider this as a two-tailed test, meaning that the critical region is split into two parts, often having equal probabilities, in each tail of the distribution of the test statistic. The alternative hypothesis ∃i,j(μi ≠ μj) can be further broken down into ∃i,j(μi < μj) or ∃i,j(μi > μj), indicating that at least one algorithm has a lower or higher average rank than the others, respectively.

ii. We have chosen a Significance level: α = 0.05

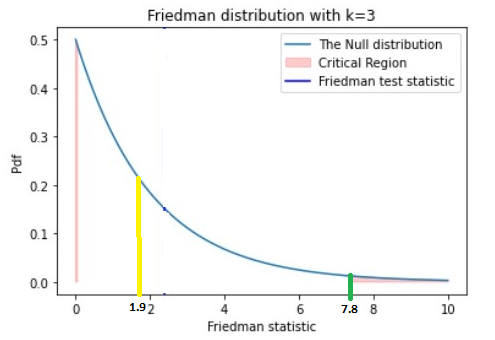
iii. Friedman's statistic F = 1.4

The Friedman's statistic, denoted as F and calculated using a specific formula, can be approximated by a chi-squared distribution with k-1 degrees of freedom.



1. Critical Values (Friedman's statistic): Number of blocks N =10, α = 0.025, k=3 is 0.0506 and 7.8
2. Based on our alternative we defined our critical region as follows,

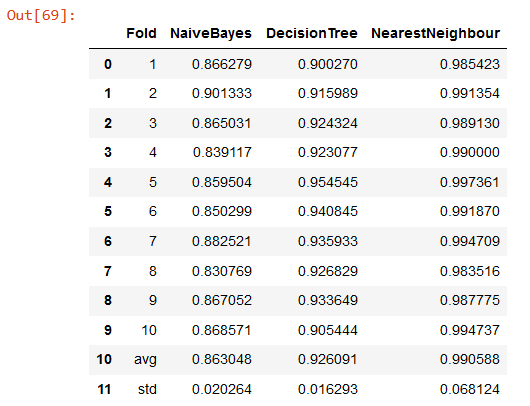
**Critical Region: Region 1: Reject H0 if F < 0.0506 or F>7.8**



vi. Conclusion: So, in our case we accept our null hypothesis (H0), because the computed Friedman Statistic= 1.4 is not in the critical region.

**The F measure Data and Rank Comparison of machine learning algorithms across multiple test folds:**

We use the sample F-measure data, assuming it comes from different data sets. The following table shows the ranks in brackets for three different machine learning algorithms (Naive Bayes, Decision Tree, Nearest Neighbor) across 10 folds:

  
Table 3: Fmeasures Cross-Validation Table (Example 12.4)

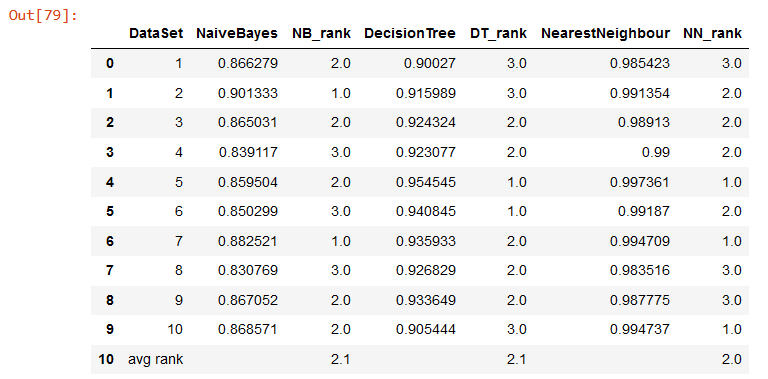


Table 4: F-measures; Rank Table (Example 12.4)

1. **Friedman Test**

i) We define the Null and Alternative Hypotheses

The null and alternative hypotheses for the Friedman Test can be written as follows:

H0: All algorithms perform equally, and the average ranks are the same.

In mathematical form:

H0: ∀i,j(μi = μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that all algorithms have the same average rank.

H1: At least one of the algorithms performs differently than the others, and the average ranks are not all equal.

In mathematical form, the null and alternative hypotheses can be expressed as:

H1: ∃i,j(μi ≠ μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that at least one algorithm has a different average rank than the others.

Since our alternative hypothesis is H1: ∃i,j(μi ≠ μj), We consider this as a two-tailed test, meaning that the critical region is split into two parts, often having equal probabilities, in each tail of the distribution of the test statistic. The alternative hypothesis ∃i,j(μi ≠ μj) can be further broken down into ∃i,j(μi < μj) or ∃i,j(μi > μj),

indicating that at least one algorithm has a lower or higher average rank than the others, respectively.

ii. We have chosen a Significance level: α = 0.05

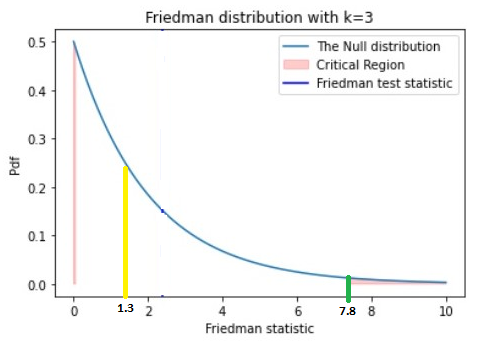
iii. Friedman's statistic F = 1.3

The Friedman's statistic, denoted as F and calculated using a specific formula, can be approximated by a chi-squared distribution with k-1 degrees of freedom.

C:\Users\Ali Abid\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1AF70049.tmp  
iv) Critical Values (Friedman's statistic): Number of blocks N =10, α = 0.025, k=3 is 0.0506 and 7.8

v)Based on our alternative we defined our critical region as follows,

**Critical Region: Region 1: Reject H0 if F < 0.0506 or F>7.8**



vi. Conclusion: So, in our case we accept our null hypothesis (H0), because the computed Friedman Statistic= 1.3 is not in the critical region. Our null hypothesis is True.

H0: All algorithms perform equally, and the average ranks are the same.

**The training time Data and Rank Comparison of machine learning algorithms across multiple test folds:**

We use the sample execution time data, assuming it comes from different data sets. The following table shows the ranks in brackets for three different machine learning algorithms (Naive Bayes, Decision Tree, Nearest Neighbor) across 10 folds:

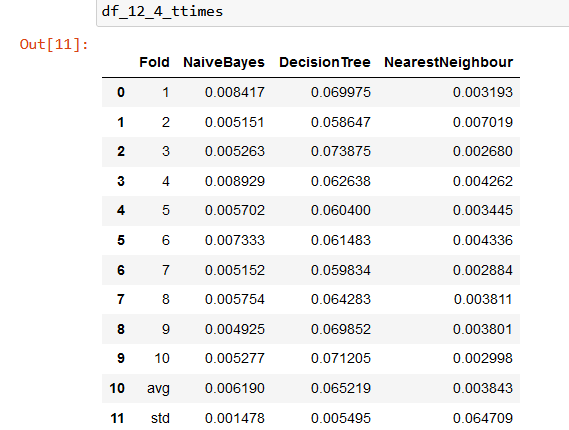


Table 5: Training Times; Cross-Validation Table (Example 12.4)

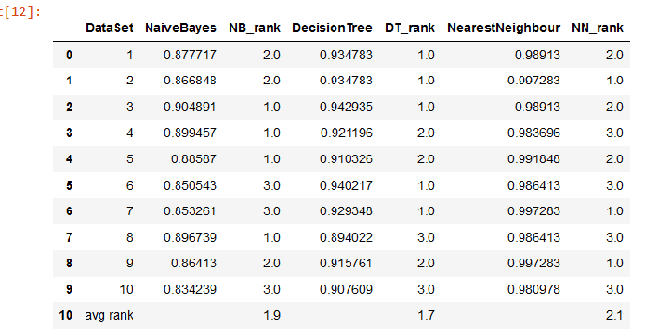


Table 5: Trainig Times; Rank Table (Example 12.4)

1. **Friedman Test**

i) We define the Null and Alternative Hypotheses

The null and alternative hypotheses for the Friedman Test can be written as follows:

H0: All algorithms perform equally, and the average ranks are the same.

In mathematical form:

H0: ∀i,j(μi = μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that all algorithms have the same average rank.

H1: At least one of the algorithms performs differently than the others, and the average ranks are not all equal.

In mathematical form, the null and alternative hypotheses can be expressed as:

H1: ∃i,j(μi ≠ μj), where μi is the average rank of algorithm i and μj is the average rank of algorithm j. This means that at least one algorithm has a different average rank than the others.

Since our alternative hypothesis is H1: ∃i,j(μi ≠ μj), We consider this as a two-tailed test, meaning that the critical region is split into two parts, often having equal probabilities, in each tail of the distribution of the test statistic. The alternative hypothesis ∃i,j(μi ≠ μj) can be further broken down into ∃i,j(μi < μj) or ∃i,j(μi > μj), indicating that at least one algorithm has a lower or higher average rank than the others, respectively.

ii. We have chosen a Significance level: α = 0.05

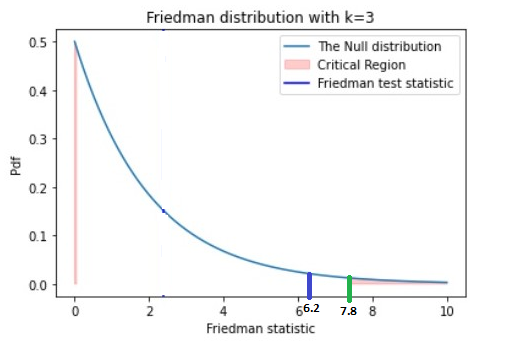
iii. Friedman's statistic F = 6.2

The Friedman's statistic, denoted as F and calculated using a specific formula, can be approximated by a chi-squared distribution with k-1 degrees of freedom.

C:\Users\Ali Abid\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\99A1339D.tmp  
iv) Critical Values (Friedman's statistic): Number of blocks N =10, α = 0.025, k=3 is 0.0506 and 7.8

v)Based on our alternative we defined our critical region as follows.

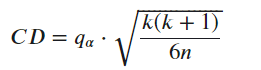
**Critical Region: Region 1: Reject H0 if F < 0.0506 or F>7.8**

****

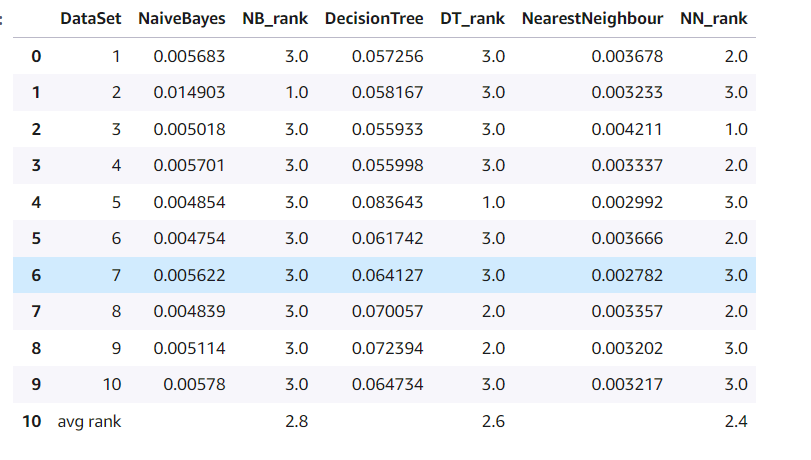
vi. Conclusion: So, in our case we accept our null hypothesis (H0), because the computed Friedman Statistic = 6.2 is not in the critical region. Our null hypothesis is True.

H0: All algorithms perform equally, and the average ranks are the same.

**The Nemenyi Test:**



The value of 𝑞𝛼 in the Nemenyi test depends on both the significance level 𝛼 and the number of algorithms, denoted as K. For Example, when 𝛼=0.05 and k = 3, the value of 𝑞𝛼 and the value of df = inf is 2.343 (studentized range distribution). In our specific example, this leads to a critical difference of 1.047.

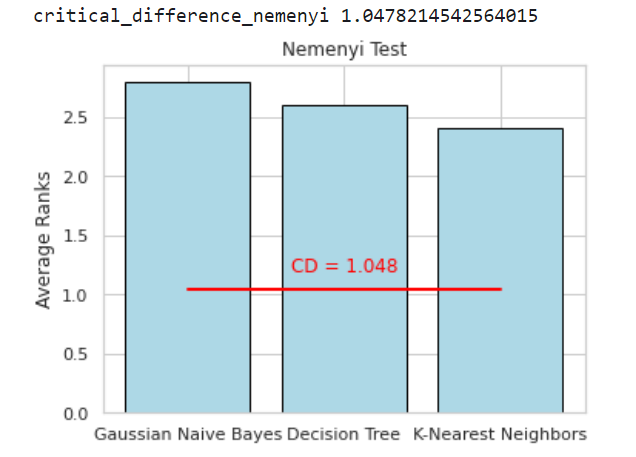


In comparison, if the average ranks were 2.8, 2.6, and 2.4:

Friedman Statistic: 11.60

Then the null hypothesis would be rejected at the significance level α = 0.05.

The Friedman test provides insights into whether there are significant variations in the overall average ranks. However, additional examination is required at a pairwise level. To accomplish this, a post-hoc test is applied once the Friedman test yields significant results. The objective is to compute the critical difference (CD) for comparing the disparity in average ranks between two algorithms. The Nemenyi test is post-hoc statistical methods used to determine the critical difference based on specific patterns in the data.



### **REFERENCES**

1. P. A. Flach, Machine learning: the art and science of algorithms that make sense of data. Cambridge: Cambridge University Press, 2012.
2. S*cipy.stats.friedmanchisquare(scipy.stats.friedmanchisquare - SciPy v1.9.3 Manual*. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.friedmanchisquare.html
3. R. E. Walpole, R. H. Myers, S. L. Myers, and Keying Ye, Probability & statistics for engineers & scientists: MyStatLab update. Boston: Pearson, 2017.