**INOFRMATICS INSTITUTE OF TECHNOLOGY**

**In collaboration with**

ROBERT GORDEN UNIVERSITY ABERDEEN

BSc. Artificial Intelligence & Data Science

Level 05

CM 2604

Machine Learning

**Coursework Report**

NADUN SHAMIKA SENARATHNE

IIT ID: 20210488

RGU ID: 2117538

Table of Contents

[Corpus Preparation 2](#_Toc130548597)

# Corpus Preparation

The UCI website, which contains a huge selection of datasets that may be utilized for any machine learning task, is where the Spambase Data Set was collected from. The last column of the dataset, which has around 4600 data and 58 attributes/features, indicates if the provided email is spam or not. A spam email is indicated by the number 1, and a non-spam email by the number 0. By examining the qualities that show if a certain character or given word is commonly occurring in a letter, the emails were divided into spam and no spam groups.

In terms of the data preprocessing, actions like data cleansing were taken. The processes included examining and eliminating null values, eliminating duplicate values, and eliminating outliers.

K-nearest neighbors and Decision Tree, two machine learning techniques, were used to classify emails as spam or not spam. Following careful management of the data cleaning phase, the dataset was split into train and test halves, with 70% of the data being designated as train data and the remaining portion as test data. It is considerably better to allocate between 20 and 30% because the data can then have a larger percentage of training. Yet it varies according on the circumstances. There is no such thing as an ideal split percentage for data training.

# Solution Methodology

KNN and Decision Tree algorithms were used to classify spam and non-spam emails. An overview of the two algorithms that were employed may be found below.

**KNN**, or the k-nearest neighbor algorithm, is a non-parametric model that locates the k data points that are closest to a test point given a scenario, and then classifies the test point based on the majority class of the k-neighbors.

**Decision Tree**,The decision tree method divides the dataset into groups that resemble tree topologies. According to the values of the inputs, the trees will have a stopping criterion.

Using the Spambase dataset, which initially divides the data into train and test sets, the KNN and Decision Tree models were employed. Then, features were employed as inputs, and labels were used to produce results. The PCA technique was used to complete the classification challenge since it enhances the models' effectiveness and performance. To select the ideal parameters and enhance the models' output, methods like grid search were used. Moreover, feature engineering was carried out since it finds the dataset's most instructive characteristics, develops new features, and captures the intricate interactions between labels.

# Evaluation Criteria

The metrics employed for the spam, no spam email categorization scenario, including as accuracy, precision, recall, and F1-scores, are described below along with the justifications for their use.

* **Accuracy**: In the case described, it is essential to have a high accuracy rate because it is crucial to properly differentiate and identify spam and non-spam emails in order to prevent missing crucial communications.
* **Precision & Recall**: For the classification instance, precision and recall metrics were utilized because they both represent the model's capacity to properly identify the number of true positives and false positives separately.
* **F1-Score**: This score was utilized to assess the model's performance since it balances precision and recall and provides a thorough picture of the models' overall effectiveness.

**Accuracy**

Shape

Description automatically generated with low confidence

**Precision**

Text

Description automatically generated with low confidence

**Recall**

**A picture containing text

Description automatically generated**

**F1-Score**

Table

Description automatically generated

# Model Evaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | KNN with PCA | Decision Tree with PCA | KNN without PCA | Decision Tree without PCA |
| Train and Test split proportion | Train – 75%,  Test – 25% | Train – 75%,  Test – 25% | Train – 70%,  Test – 30% | Train – 70%,  Test – 30% |
| Test Accuracy | 90.71% | 86.31 % | 90.13 % | 91.58 % |
| Precision (weighted) | 91% | 86% | 90% | 92% |
| Recall (weighted) | 91% | 86% | 90% | 92% |
| F1-Score (weighted) | 91% | 86% | 90% | 92% |

## Confusion matrix for KNN (with PCA):

Chart, treemap chart

Description automatically generated

## Classification Report for KNN (PCA):

## Calendar Description automatically generated with medium confidence

## 

## Confusion matrix for Decision Tree (with PCA):

Chart, treemap chart

Description automatically generated

## Classification Report for Decision Tree (PCA):

## 

## Confusion matrix for KNN (without PCA):

Chart, treemap chart

Description automatically generated

## Classification Report for KNN (without PCA):

## 

Calendar

Description automatically generated

## Confusion matrix for Decision Tree (without PCA):

Chart, treemap chart

Description automatically generated

## 

## Classification Report for Decision Tree (without PCA):

## 

Calendar

Description automatically generated

Based on the metrices that were utilized, both models gave good scores of accuracies. When PCA was applied, KNN, however, outperformed Decision Tree. By having a walkthrough classification report it can be argued that a high precision can be obtained if the amount of false positives are taken into consideration and high recall can be obtained if the false negatives were taken into consideration.

# Experimental Results

Using the K-Nearest and Decision tree algorithms, an experiment was done to evaluate and categorize the results of spam and no spam emails. More than 4800 emails made up the Spambase dataset, which was split into 25% of the data for testing and 75% of the data for training.

In K-Nearest methods, the number of k components was set automatically to determine the Euclidian distance between each data point from the hyperparameter tuning by passing its value. To reduce the problems associated with overfitting, the decision tree's tree depth was also passed the parameters associated with it.

To improve the performance of the two models, various metrics were applied to the training data. According to the findings, KNN had an accuracy rate of 90.71% after PCA was applied, whereas decision tree had an accuracy rate of 86.31 %.

Overall, it was found that KNN was more effective than decision tree when PCA was used, but decision tree was more effective when PCA was not used.

# Limitations & Further Enhancements

## Limitations:

* As the dataset used to train the model does not contain any real-world data, data bias may have an impact on the model's performance.
* A decision tree has a potential of being overfitted, especially if it is too complicated, the data is noisy, or the training process was improper.

## Ways to overcome the limitations:

* Utilizing a more representative dataset to provide results that are more exact and accurate.
* Techniques like pruning can be used for decision tree classification to address the overfitting issue.

## Future enhancements:

* Testing out various methods, such as Naive Bayes and Support Vector Machines (SVM), and comparing their performance to that of Decision Tree and K-Nearest Neighbors algorithms.
* Using the same models to various tasks, like sentiment analysis and topic classification, and evaluating the outcomes.

# GitHub project URL:

The whole code that was used to categorize spam and non-spam email using KNN and Decision Tree classifiers is available at the GitHub link below.

GitHub Logos and Usage · GitHub

[github.com/Nadun999](https://github.com/Nadun999/spam-email-classification-using-KNN-DT)

# Appendix: