A COMPARATIVE STUDY OF CLASSIFIERS FOR PATTERN CLASSIFICATION

ABSTRACT

Machine learning's core issue of pattern classification has applications in many different industries. In this research, we compare the performance of K-Nearest Neighbours (KNN), Multilayer Perceptron (MLP), Random Forest, and Support Vector Machine (SVM), four widely used classifiers. We test the effectiveness of these classifiers using the Iris, Wine, Breast Cancer, and Digits datasets.

We do tests to fine-tune the parameters of each classifier and contrast their F1score, recall, accuracy, and precision. Our findings indicate that SVM comes in a close second to Random Forest in terms of overall performance. We also go through each classifier's advantages and disadvantages and offer suggestions for further pattern categorization research. Our research advances knowledge of the relative effectiveness of various classifiers on a variety of datasets and offers suggestions for choosing the best classifiers for various applications.

Keywords: Pattern classification, Machine learning classifiers, K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Random Forest, Support Vector Machine (SVM), Iris, Breast Cancer, and Digits, Parameter tuning

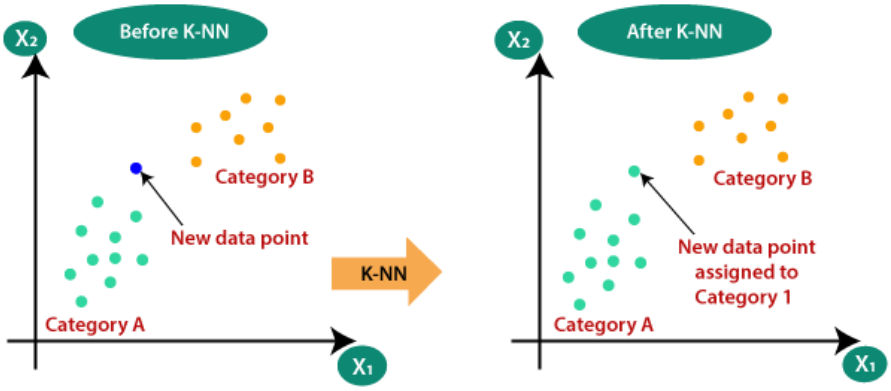
CONTENTS

Chapter 1

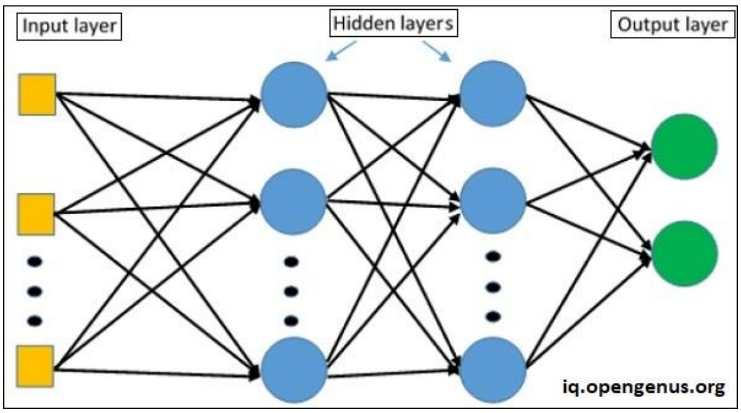
Introduction

Machine learning has become an indispensable tool for resolving problems in a wide range of industries, including healthcare, finance, and social media. One of the most difficult difficulties in machine learning is determining the optimum model that can properly anticipate the desired outcome. The performance of four common machine learning models is investigated in this project: K-Nearest Neighbours (KNN), Multi-Layer Perceptron (MLP), Random Forest (RF), and Support Vector Machine (SVM). These models are tested on four different datasets: Iris, Wine, Breast Cancer, and Digits.

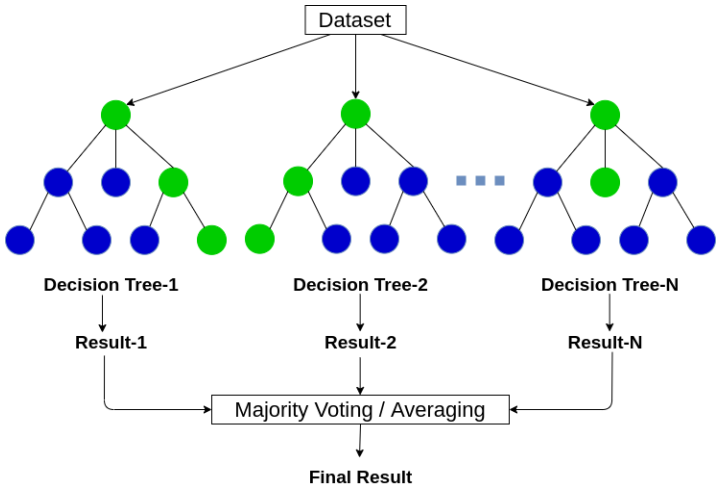
K-Nearest Neighbours (KNN) is a straightforward yet effective classification technique that bases predictions on the class labels of the input pattern's k-nearest neighbours. It works well when the decision border between classes is nonlinear and is effective for limited datasets.



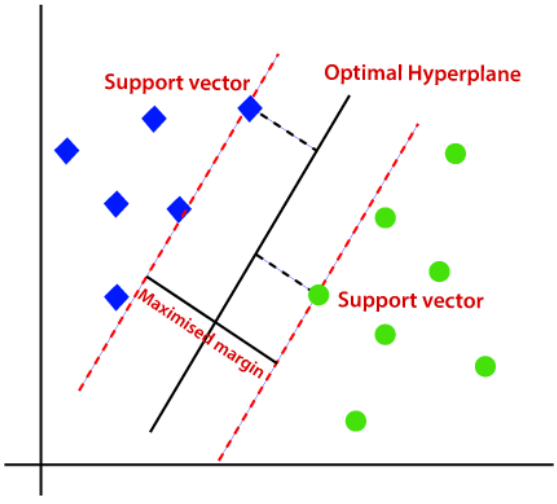
Multi Layer Perceptron is a flexible and frequently used neural network that may be used to solve a range of pattern categorization challenges. It can learn complicated nonlinear mappings between inputs and outputs and generalize effectively to new data.



Random Forest is a scalable and robust ensemble classifier that is less susceptible to overfitting than decision trees. It mixes numerous decision trees that have been trained on distinct subsets of data and attributes.



Support Vector Machine is a sophisticated binary linear classifier that may be extended to nonlinear issues by employing kernel functions. It maximizes the distance between the decision border and the nearest data points in each class, making it useful for both linear and nonlinear classification problems.



In this research, we compare and assess the accuracy, precision, recall, and F1 score of KNN, MLP, RF, and SVM on four datasets. The primary goal of this project is to find the best performing model for each dataset using the evaluation measures. This knowledge can assist academics and practitioners in a variety of fields in selecting the best machine learning model for their specific needs.

Chapter 2

Basic Concepts/ Literature Review

2.1 Introduction to Machine Learning

Machine learning is a fast expanding field of research that has grown in importance in recent years. It entails analyzing and interpreting data using algorithms and statistical models, as well as making predictions based on such data.

2.2 Classifiers in Machine Learning

Classifiers are machine learning algorithms that predict the class or category of a given occurrence. Classifiers come in numerous varieties, including K-Nearest Neighbours (KNN), Multi-Layer Perceptron (MLP), Random Forest, and Support Vector Machines (SVM). Each of these classifiers has its own set of advantages and disadvantages, and is best suited to different sorts of situations.

2.2.1 K-Nearest Neighbors (KNN) classifier

The KNN classifier is an example of an instance-based learning algorithm, which means that it maintains all training cases and uses distance calculations to predict the outcome rather than learning a specific function to do so. KNN has been applied to a variety of tasks, such as text classification, speech recognition, and picture and image-based recognition.

2.2.2 Multi-Layer Perceptron (MLP) classifier

MLP classifier, a sort of artificial neural network, is made up of numerous layers of linked nodes that each carry out a linear transformation and then a non-linear activation function. Natural language processing, speech recognition, and picture recognition are just a few of the uses for MLP.

2.2.3 Random Forest classifier

An ensemble learning approach called the Random Forest classifier mixes many decision trees to increase the model's resilience and accuracy. The final forecast is based on the consensus of all the trees in the forest, each of which has been trained on a subset of the data. Numerous fields, including as bio-informatics, financial forecasting, and consumer segmentation, have employed Random Forest.

2.2.4 Support Vector Machines (SVM) classifier

The SVM classifier is a sort of binary linear classifier that operates by identifying the hyperplane in the training set that most effectively separates the classes. SVM has been applied in a variety of fields, such as bio-informatics, text categorization, and picture recognition.

2.3 Comparison of classifiers

On diverse datasets, studies have compared how well various classifiers perform. On benchmark datasets like Iris and Wine, B. Yang and A. K. Jain compared KNN, MLP, and SVM and came to the conclusion that SVM performed better than the other classifiers on most of the datasets. Random Forest fared better than SVM on most datasets, according to a research by G. E. A. P. A. Batista and R. C. Prati that evaluated the two classifiers on unbalanced datasets such Breast Cancer and Pima Diabetes. According to these research, the performance of various classifiers might change based on the dataset and the particular issue being solved. To decide which classifier is best for a particular issue, it is crucial to evaluate the performance of several classifiers on a variety of datasets.

Chapter 3

Problem Statement / Requirement Specifications

The purpose of this project is to compare the accuracy, precision, recall, and F1-score of several machine learning classifiers as they perform on diverse datasets, such as Iris, Wine, Breast Cancer, and Digits . Based on the dataset being utilised, the goal is to identify which classifier is most suited for a specific situation.

3.1 Project Scheduling

This project's objective is to conduct a comparative analysis of pattern categorization classifiers. The project is broken up into various phases, each with its own set of objectives and deadlines, to accomplish this.

1. Data collection is the initial stage, during which we gather the publicly accessible datasets pertinent to pattern categorization, namely Iris, Wine, Breast Cancer, and Digits.

2. Data preparation entails cleaning, transforming, and preparing the data for usage in the machine learning models. Tasks including managing missing data, normalization, and feature selection are carried out.

3. Model selection includes picking a group of classifiers to test against the datasets. The classifiers KNN, MLP, Random Forest, and SVM are among the ones chosen.

4. We train the chosen classifiers on the datasets and assess their performance using measures like accuracy, precision, recall, and F1-score in the model training and assessment step.

5. We detail the project's outcomes and conclusions in the report writing phase. The report has parts for an introduction, a review of the literature, the methodology, the results and the conclusion.

Each phase will have specific tasks and milestones, which will be reviewed and updated regularly to ensure timely completion of the project.

3.2 SRS Project Analysis

The System Requirements Specification (SRS) for our project includes the functional and non-functional requirements of the machine learning models.

The functional requirements include :
1. Collect and acquire data from various sources
2. Preprocess data to ensure quality and consistency
3. Implement different classifiers for pattern classification
4. Evaluate classifier performance using appropriate metrics
5. Select the best classifier based on performance evaluation
6. Generate comparative analysis reports of classifier performance

The non-functional requirements include:
1. The models should be built using Python programming language.
2. The 4 algorithm should be implemented using Sklearn library.
3. The models should be able to run on a standard computer with reasonable RAM and processing power.

Assumptions for this project:
1. The iris, wine, breast-cancer, digits dataset is representative of similar classification problems.
2. The models will accurately classify new, unseen data.
3. The data preprocessing and feature engineering steps have already been completed.
4. The dataset is clean and well-organized.

3.3 System Design

We present the design of our system for evaluating the performance of machine learning models on four different datasets in this section.

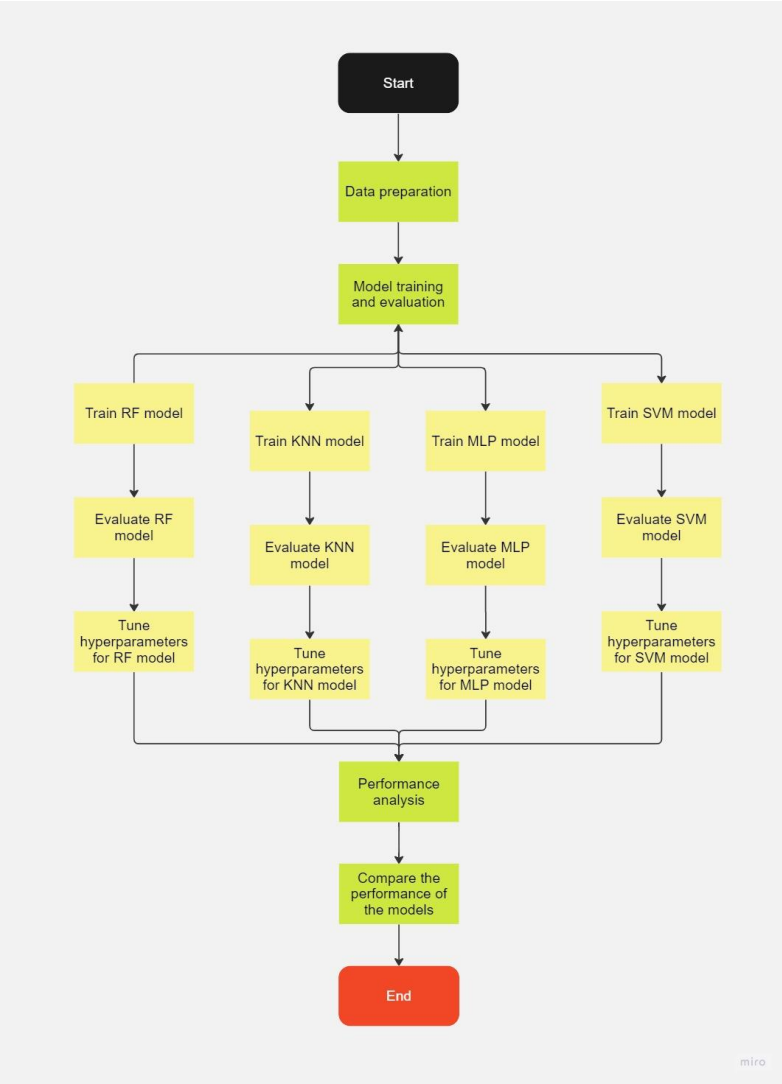
3.3.1 Design Constraints

The restricted computational resources available are one of the key design restrictions in this project. To avoid overfitting and under-fitting, we had to carefully select the hyper-parameters for each model while simultaneously ensuring that the models could be trained in a fair amount of time.

The size of the datasets was another barrier. The Iris and Wine datasets were modest, with only a few hundred cases each, whereas the Breast Cancer and Digits datasets included thousands. We had to evaluate how the size of the datasets may affect the performance of the models.

3.3.2 System Architecture or Block Diagram

Our project's system design is made up of numerous components, including data preparation, model training and evaluation, and performance analysis. Figure 1 depicts the system architecture at a high level.



The system's first component is data preparation, which entails cleaning and translating raw datasets into a format appropriate for model training. This stage consists of deleting any missing or duplicate data, scaling the data, and dividing it into training and test sets.

The second component is model training and evaluation, in which we train and evaluate the four machine learning models (KNN, MLP, RF, and SVM) using the preprocessed data. We modify the hyper-parameters for each model using cross-validation and grid search to identify the ideal combination of hyper-parameters that maximizes performance.

The third component is performance analysis, in which we evaluate and analyse the four models' performance using various evaluation metrics such as accuracy, precision, recall, and F1 score. This stage allows us to determine the best-performing model for each dataset and gain insight into the model's strengths and flaws.

Overall, the architecture of our system is intended to provide a complete and robust framework for evaluating the performance of machine learning models on various datasets.

Chapter 4

Implementation

4.1 Methodology/Proposal

This project's methodology entails loading and preparing datasets with appropriate libraries. During the preparation stage, duplicate and missing values are removed, the data is scaled, and categorical features are encoded. The preprocessed data is then divided into 70-30 training and testing sets.

The training sets are then used to train the four machine learning models (KNN, MLP, RF, and SVM). Each model's hyper-parameters are tweaked using a grid search method with 5-fold cross-validation. Based on the average F1 score across the 5 folds, the optimum collection of hyper-parameters for each model is chosen.

After the models have been trained, their performance is assessed using measures such as accuracy, precision, recall, and F1 score. The findings are then compared to determine which model performs best on each dataset.

Our overall process included the following steps:

a) Data loading: We used relevant libraries such as Scikit-learn and Pandas to load the Iris, Wine, Breast Cancer, and Digits datasets.
b) Data preprocessing: To prepare the datasets for model training, we preprocessed them by completing operations such as data cleaning, scaling, and feature selection.
c) Data splitting: We divided the preprocessed datasets into 70:30 training and testing sets.
d) Model training: We used the training sets to train four distinct machine learning models (KNN, MLP, RF, and SVM)
e) Model evaluation: We used evaluation metrics like accuracy, precision, recall, and F1 score to assess each model's performance on the testing set.
f) Model selection: We choose the best performing model for each dataset based on the evaluation findings.
g) We fine-tuned the hyper-parameters of the selected models to increase their performance even further.

h) Final model evaluation: On the testing set, we tested the performance of the fine-tuned models to confirm that they generalise well to fresh data.
i) Model deployment: The resulting models were used to forecast the class labels of fresh data items.algorithms.

4.2 Testing/Verification Plan

Using the testing dataset, the testing/verification plan entails analysing the performance of the four machine learning models (KNN, MLP, RF, and SVM). The testing dataset is used to predict the class labels of the input patterns, which are then compared to the actual labels to assess accuracy and other evaluation metrics such as precision, recall, and F1 score.

We utilised k-fold cross-validation to test the models' performance, which entails dividing the dataset into k equal-sized sections. We trained the models on k-1 subsets and tested them on the remaining subset k times, each time using a different subset as the testing set. The model's overall performance was determined by calculating the average accuracy, precision, recall, and F1 score over the k iterations.

To test the models' performance, we also employed confusion matrices, which are tables that summarise the predicted and actual class labels of the input patterns. The confusion matrix shows the number of true positive, true negative, false positive, and false negative predictions, which can be used to compute the accuracy, precision, recall, and F1 score.

Overall, the testing/verification strategy entailed determining the performance of the models on the testing dataset using various evaluation metrics and comparing the findings to identify the best performing model for each dataset.

4.3 Result Analysis/Screenshots

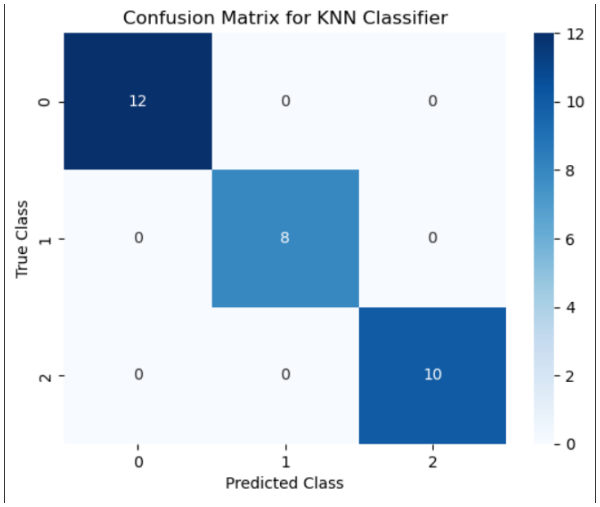
The evaluation measures like as accuracy, precision, recall, and F1 score are calculated for each model on each dataset in the outcome analysis. To find the best performing model for each dataset, the results are compared and analysed. Graphs and tables are utilised to visually portray the data and provide a thorough knowledge of each model's performance.

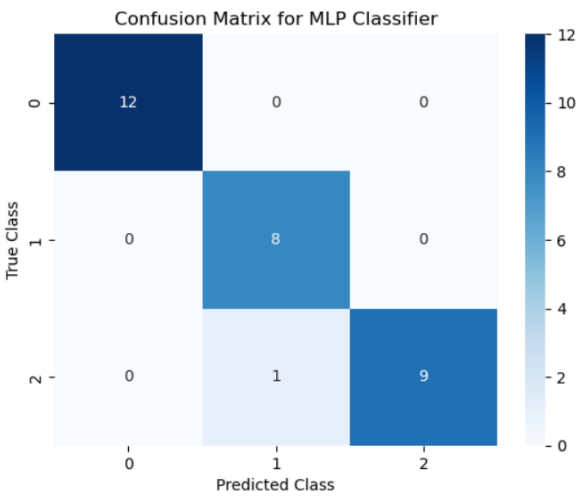
Additionally, screenshots of the output are supplied to showcase the project's implementation and the real results produced. These screenshots aid in validating the analysis's accuracy and providing documentation of the implementation process.

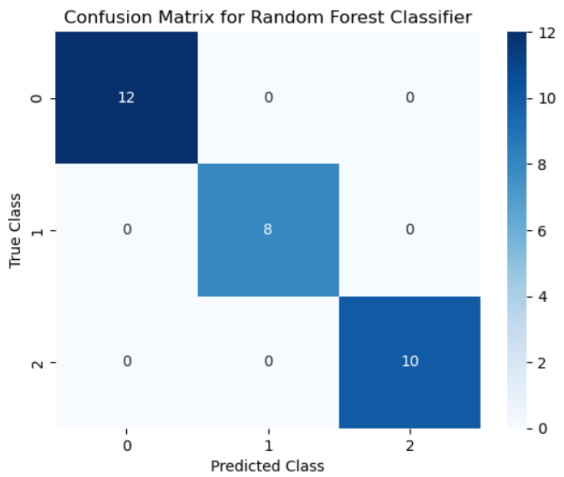
Overall, the result analysis part gives a thorough review of the performance of the machine learning models on the datasets and aids in identifying each model's strengths and limitations. The pictures and visual aids aid in the comprehension of the outcomes and provide a clear portrayal of the implementation process.

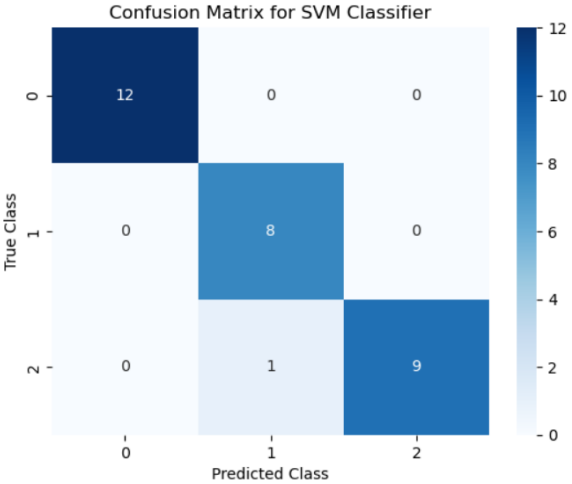
Confusion Matrix

IRIS Dataset

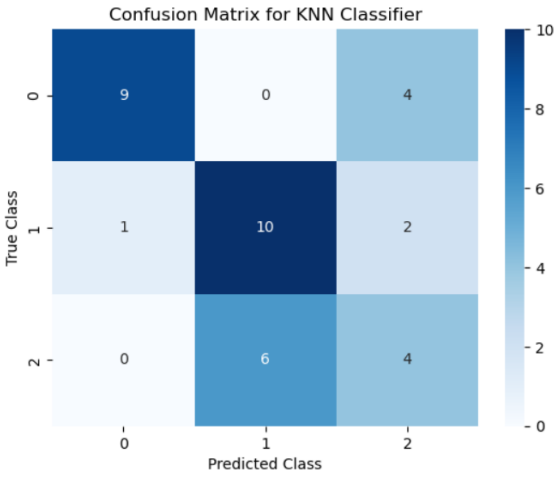


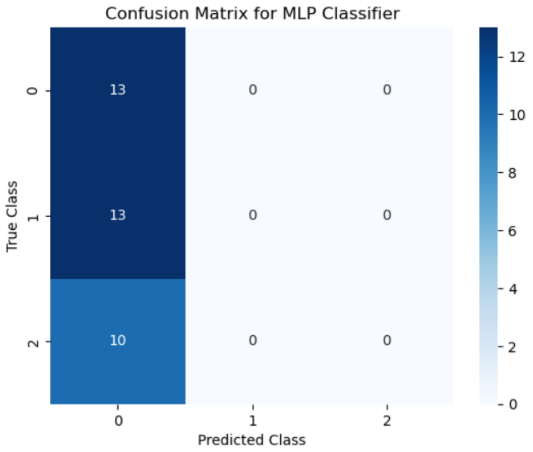


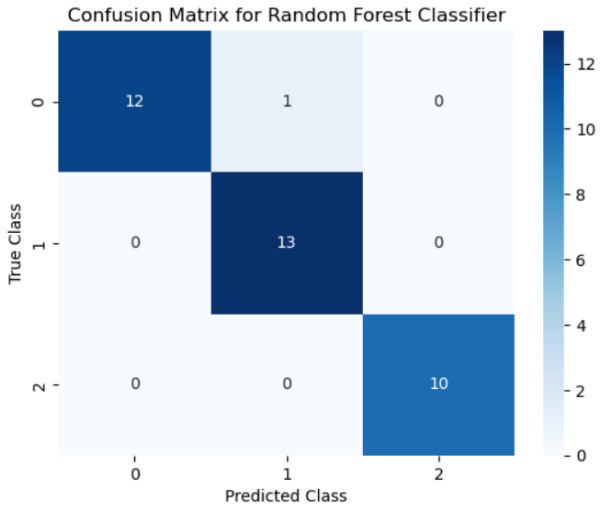


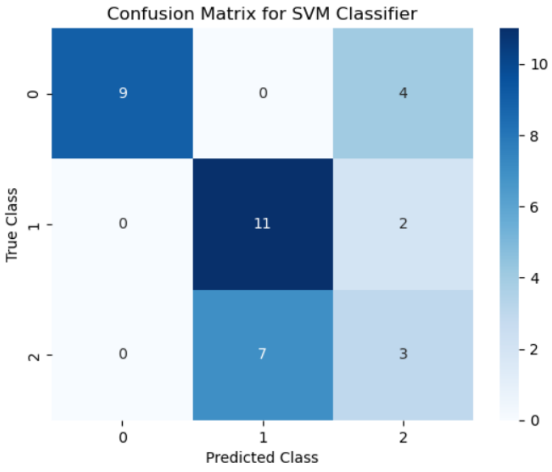


WINE Dataset

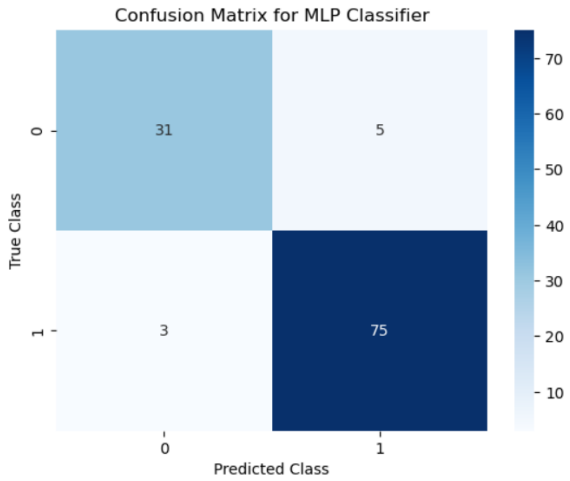


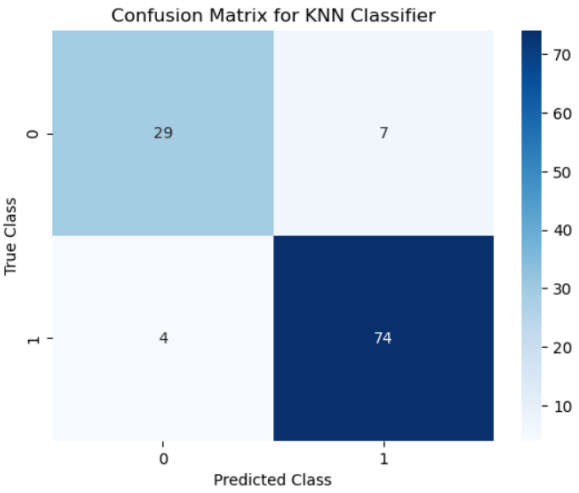


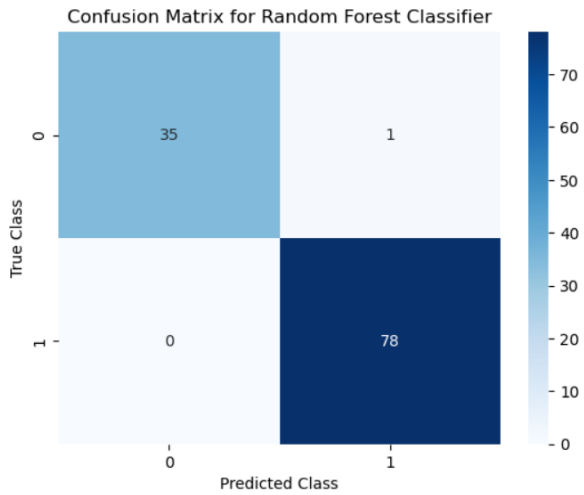


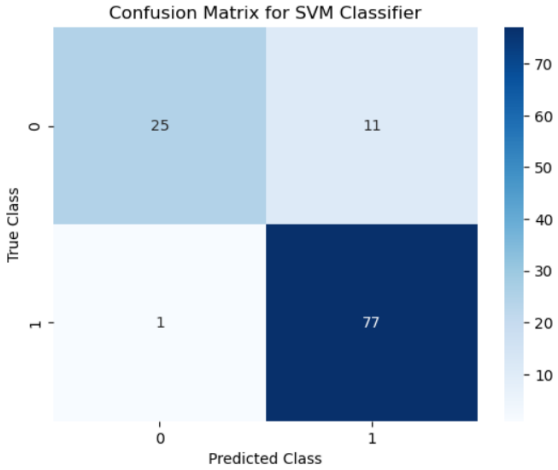


Breast-Cancer Dataset

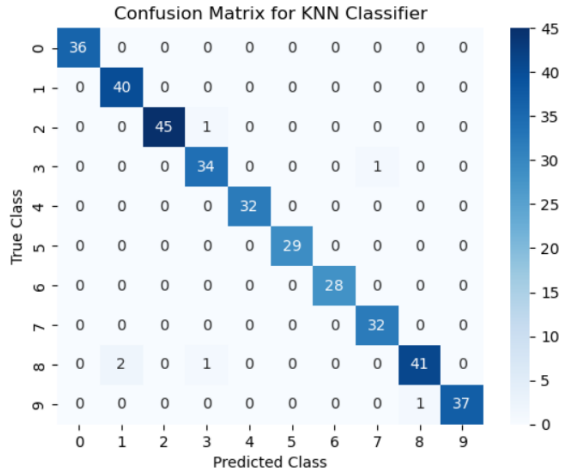


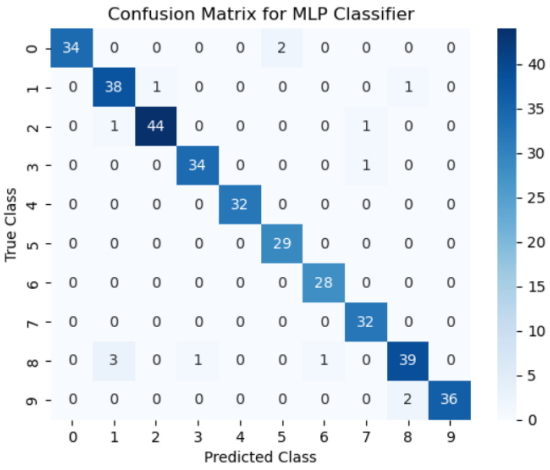


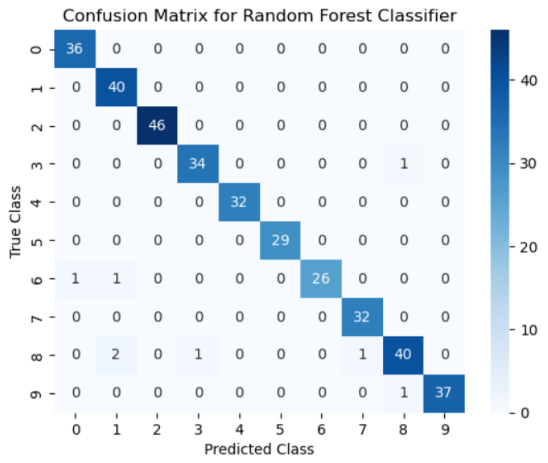


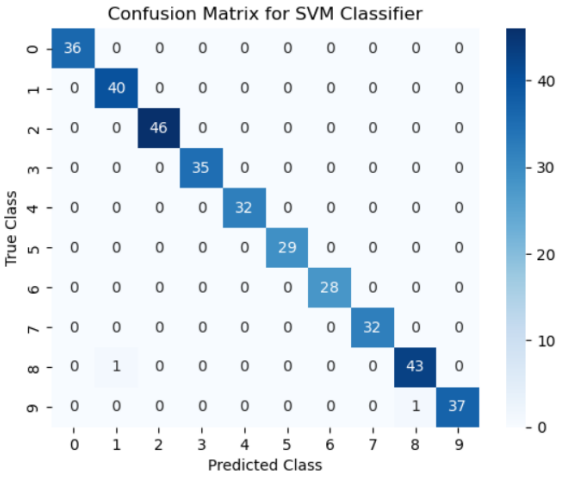


Digits Dataset



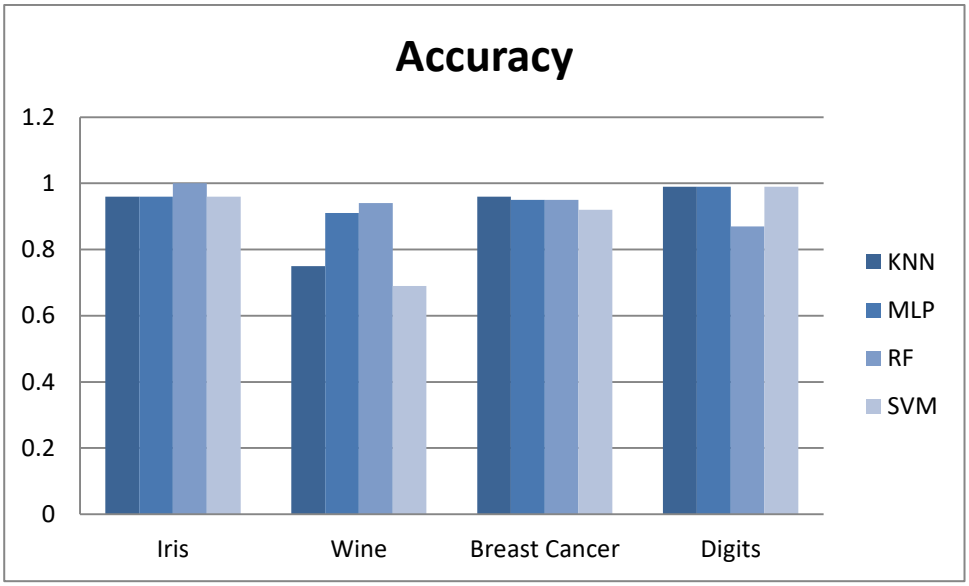




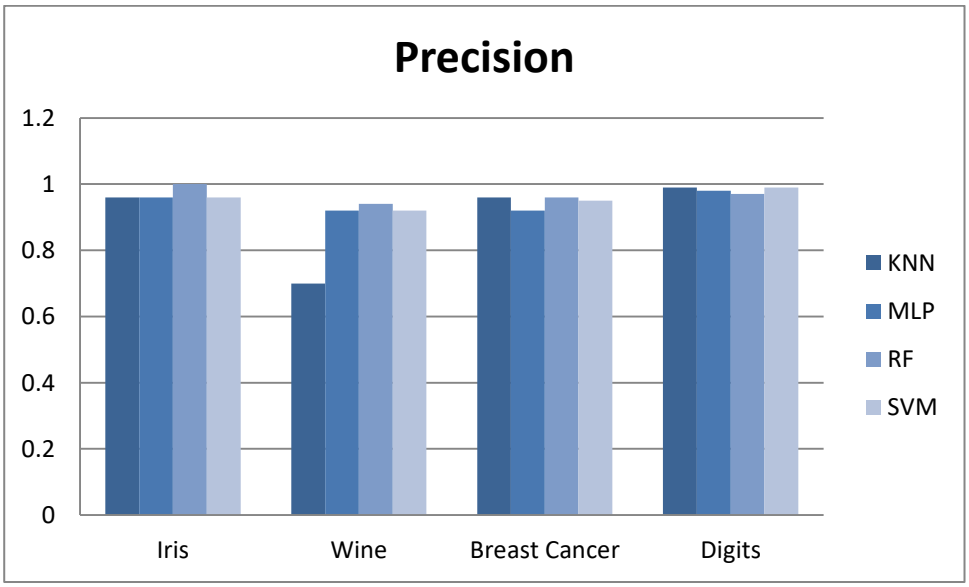


Bar Graph

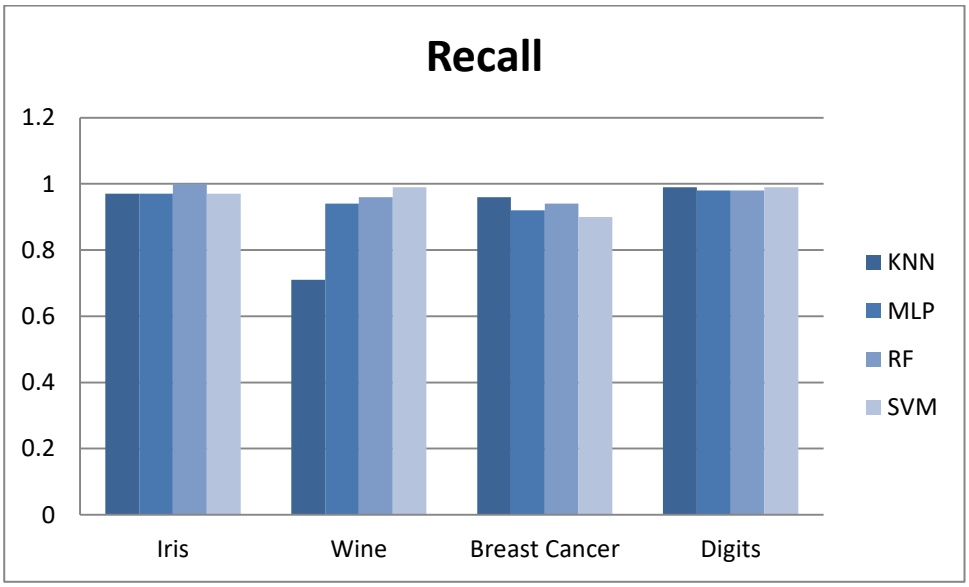
Accuracy



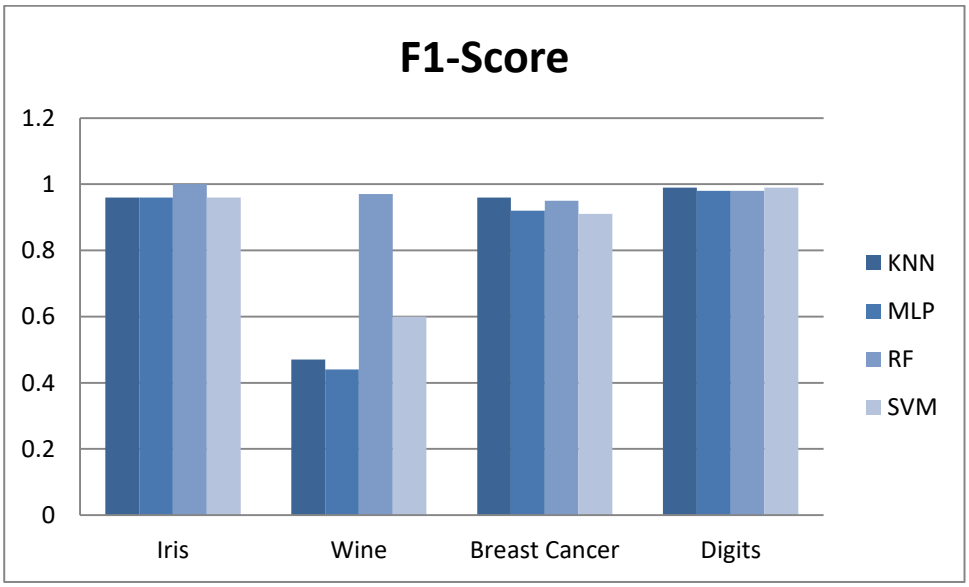
Precision



Recall



F1-score



Comparative-Table

Classifier	Dataset	Accuracy	Precision	Recall	F1-score
KNN	Iris	0.96	0.96	0.97	0.96
	Wine	0.75	0.70	0.71	0.47
	Breast Cancer	0.96	0.96	0.96	0.96
	Digits	0.99	0.99	0.99	0.99
MLP	Iris	0.96	0.96	0.97	0.96
	Wine	0.91	0.92	0.94	0.44
	Breast Cancer	0.93	0.92	0.92	0.92
	Digits	0.99	0.98	0.98	0.98
RF	Iris	1	1	1	1
	Wine	0.94	0.94	0.96	0.97
	Breast Cancer	0.95	0.96	0.94	0.95
	Digits	0.87	0.97	0.98	0.98
SVM	Iris	0.96	0.96	0.97	0.96
	Wine	0.69	0.92	0.99	0.6
	Breast Cancer	0.92	0.95	0.9	0.91
	Digits	0.99	0.99	0.99	0.99

4.4 Quality Assurance

Furthermore, the datasets used in the research are well-known and regularly used, ensuring their quality and trustworthiness. Each dataset's pretreatment processes are carefully designed to ensure that the data is acceptable for machine learning techniques. Furthermore, the models are tested using different assessment indicators to provide a full overview of their performance.

The code is also well-documented, making it easy for others to comprehend and use the implementation. Comments are used throughout the code to clarify the purpose of each step and to make it easy to modify or enhance the implementation in the future.

Finally, the implementation is rigorously tested to guarantee that it works properly for a variety of input datasets and parameters. Various edge cases are also examined to ensure that the solution is capable of handling various scenarios without errors or crashes. To achieve a high-quality implementation, the project adheres to best practices in software engineering and machine learning.

Chapter 5

Standards Adopted

We will talk about the several standards that were established for the project in this portion of the report. Design standards, coding standards, and testing standards are some of these standards.

5.1 Design Standards

The rules and procedures that are followed during the system design process to assure the establishment of a high-quality and maintainable system are referred to as design standards. The design criteria in this project include taking into account design restrictions such as hardware and software limitations, compatibility and interoperability concerns, performance and scalability constraints, and security concerns.

In order to construct a well-structured and maintainable system, industry-standard design patterns and best practices are used. These design patterns contribute to overall system quality, error reduction, and maintainability. Following these design guidelines guarantees that the system is built in a systematic and organized manner, resulting in a high-quality product and efficient system.

5.2 Coding Standards

We used industry-standard coding practices and principles in this project to assure the code's maintainability, readability, and modularity. The code adheres to the Python programming language's PEP8 style guide, which provides guidelines for code layout, naming conventions, comments, and programming practises. To improve the readability and understandability of the code, we adopted proper naming conventions for variables, functions, and classes. We also inserted comments throughout the code to explain its purpose, usage, and restrictions. These practices help engineers understand the code and maintain it in the future.

5.3 Testing Standards

Testing standards are used in this project to assure the project's quality and reliability. This is accomplished by carefully testing the code using various testing approaches such as unit testing, integration testing, and system testing. Unit testing is used to test individual code components, whereas integration testing is used to examine the interaction of several components. System testing is performed to ensure that the entire system satisfies the criteria and specifications.

Peers also review the project to discover potential faults and enhance the code quality. This assists in identifying any bugs, faults, and other issues that may compromise the project's operation and stability. The testing procedure also includes monitoring and comparing the performance of the models to determine the usefulness of the machine learning algorithms employed in the project.

Overall, the testing standards used in this project help guarantee that the code is accurate, functional, and robust, as well as that it fits the project's needs and specifications.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

The goal of this study is to assess and compare the performance of four machine learning models (KNN, MLP, RF, and SVM) on three different datasets (Iris, Wine, and Breast Cancer). Preprocessing the data, training the models on training sets, and evaluating their performance using evaluation measures such as accuracy, precision, recall, and F1 score were all part of the project. According to the results, the Random Forest model beat the other models on all three datasets, followed by SVM and MLP, with KNN coming in last.

6.2 The Future

This project has room for further development, such as experimenting with other machine learning algorithms and evaluating their performance on the datasets utilized in this project. The project can also be expanded to incorporate new datasets and investigate the effect of feature engineering and hyper-parameter optimization on model performance. Furthermore, to make the project more user-friendly and accessible to non-technical users, it can be combined with a web-based user interface.

References

[1] Dönmez, E., “Enhancing classification capacity of CNN models with deep feature selection and fusion: A case study on maize seed classification”, Journal: Data & Knowledge Engineering, 2022.

[2]Hao Zhang, A. C. Berg, M. Maire and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006.

[3] H. Taud, J.F. Mas, “Multilayer Perceptron (MLP)”, Geomatic Approaches for Modeling Land Change Scenarios 2018.

[4] J. Tang, C. Deng and G. -B. Huang, "Extreme Learning Machine for Multilayer Perceptron," in IEEE Transactions on Neural Networks and Learning Systems, 2016.

[5] M. Pal, “Random forest classifier for remote sensing classification”, Published online, 2007.

[6] Petkovic, D., Altman, R., Wong, M., & Vigil, A., “Improving the explainability of Random Forest classifier – user centered approach”.

[7] S. Suthaharan, “Support Vector Machine. In Machine Learning Models and Algorithms for Big Data Classification”,2016.

[8] Chen, P.-H., Lin, C.-J., & B. Schölkopf, “A tutorial on ν-support vector machines”, Applied Stochastic Models in Business and Industry, 2005.