Letter Recognition using SVM, KNN, and Decision Trees

Ujjawal Saini

Guru Gobind Singh Indraprastha University

B.Tech AIML

Instructor*: Prof. Amit Choudhary*

Date: 20-06-2023

Abstract

Optical character recognition, document processing, and text analysis are just a few of the practical uses for letter recognition, a fundamental problem in pattern recognition. With an emphasis on Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees, this research project intends to develop and compare several machine learning methods for letter recognition.

The dataset used for this project is the Letter Recognition dataset obtained from the UCI Machine Learning Repository. It consists of 20,000 samples of 26 uppercase letters of the English alphabet. Each sample is represented by 16 numerical attributes that capture various properties of the letter's shape.

The proposed methodology involves several key steps. First, the dataset is preprocessed to ensure data quality and prepare it for model training. Then, three different machine learning

algorithms, namely SVM, KNN, and Decision Trees, are implemented using the scikit-learn library in Python.

The performance of each algorithm is evaluated using various metrics, such as accuracy, precision, recall, and F1-score. Additionally, confusion matrices are generated to gain insights into the classification performance for each letter.

The experimental results demonstrate that all three algorithms achieve promising results for letter recognition. SVM achieves an accuracy of 93.05%, while KNN achieves an accuracy of 95.2%, and Decision Trees achieve an accuracy of 88.15%. The highest accuracy is achieved by KNN, indicating its effectiveness in handling this classification task.

In conclusion, this research project successfully explores and compares the performance of SVM, KNN, and Decision Trees for letter recognition. The results indicate that machine learning algorithms can effectively recognize letters from their numerical representations. These findings can be utilized in various applications, such as optical character recognition systems, document processing, and automated text analysis.

Keywords: letter recognition, machine learning, classification, SVM, KNN, Decision Trees

1. Introduction:

Pattern recognition's fundamental problem, letter recognition, has important applications across a wide range of industries. In processes like optical character recognition, document processing, handwriting recognition, and text analysis, the capacity to automatically recognise and categorise

characters is essential. We can efficiently extract data, convert text to speech, and create intelligent document management systems by precisely recognising letters.

This project's main goal is to create and assess several machine learning algorithms for letter recognition. We seek to enhance the precision and effectiveness of letter classification through the application of cutting-edge pattern recognition techniques. The demand for reliable and effective systems that can automatically process significant amounts of text input is the driving force behind this research endeavour.

The importance of letter recognition comes from the variety of uses it has. For example, effective letter recognition is essential in optical character recognition systems in order to turn scanned documents or photos into editable and searchable text. Furthermore, precise letter identification can improve the accuracy and readability of outcomes in text analysis tasks like sentiment analysis or topic modelling.

Letter recognition issues have shown considerable potential for machine learning methods like Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Decision Trees. These algorithms are efficient at identifying relationships and patterns in the data, which enables them to generate precise predictions and classifications.

In order to compare these algorithms' accuracy, resilience, and computational efficiency, this study attempts to investigate their capabilities. We can determine the best algorithm for letter recognition tasks in various contexts by understanding their strengths and shortcomings.

The project's approach, including data pretreatment, algorithm implementation, and evaluation, will be covered in detail in the parts that follow. The findings and conversations will provide information on how each algorithm performs and insights into how well it recognises letters. Finally, we will wrap up the project with a summary of our results, suggestions for the future, and areas that might benefit from development.

By improving our knowledge of letter recognition methods and their real-world uses, we hope to advance the field of pattern recognition through this research effort. We want to build accurate and efficient systems that can automate letter recognition activities and boost the performance of numerous text-related applications by utilising the power of machine learning.

2. Proposed Methodology:

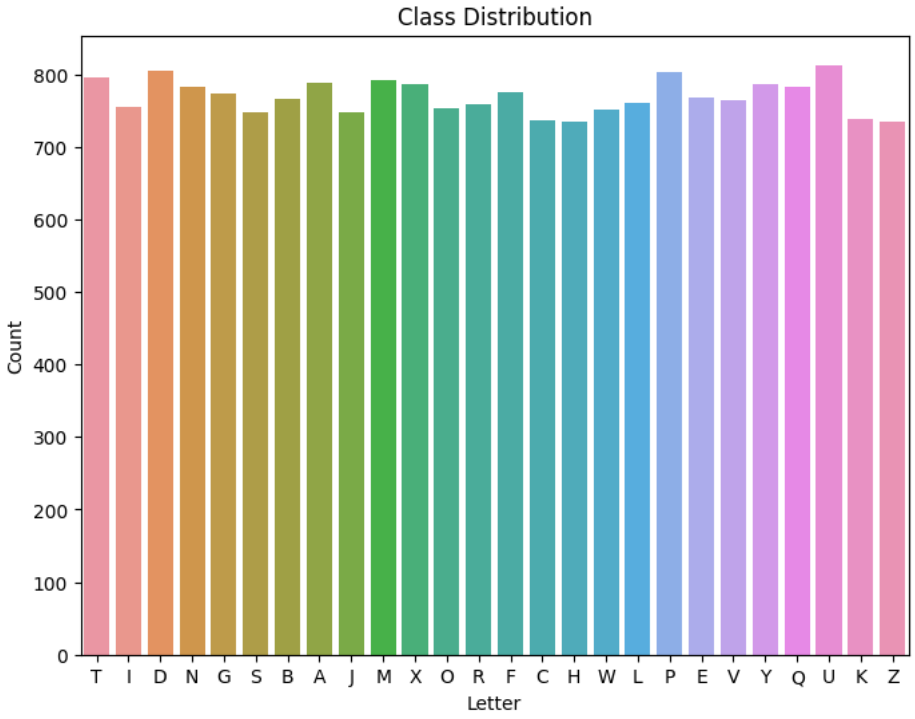
We will go over the approach used for the Letter Recognition project in this section. As part of this, three classification algorithms—Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Decision Trees—were implemented and evaluated, along with an overview of the datasets utilised, preprocessing techniques used on the data, and an overview of the datasets themselves.

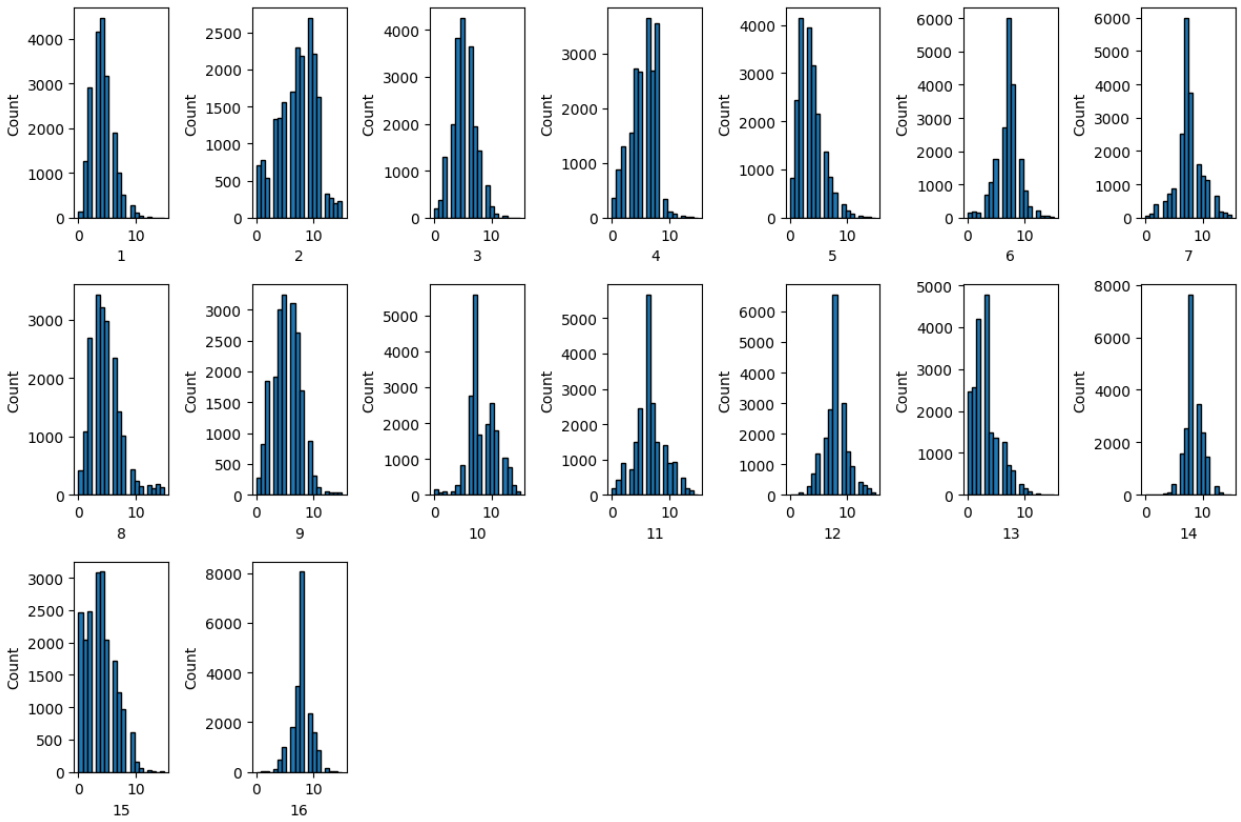
2.1 Datasets:

The dataset used for the Letter Recognition project is obtained from the UCI Machine Learning Repository [1]. It consists of 20,000 samples, where each sample represents a grayscale image of a letter of the alphabet. The dataset contains 16 attributes representing various statistical and

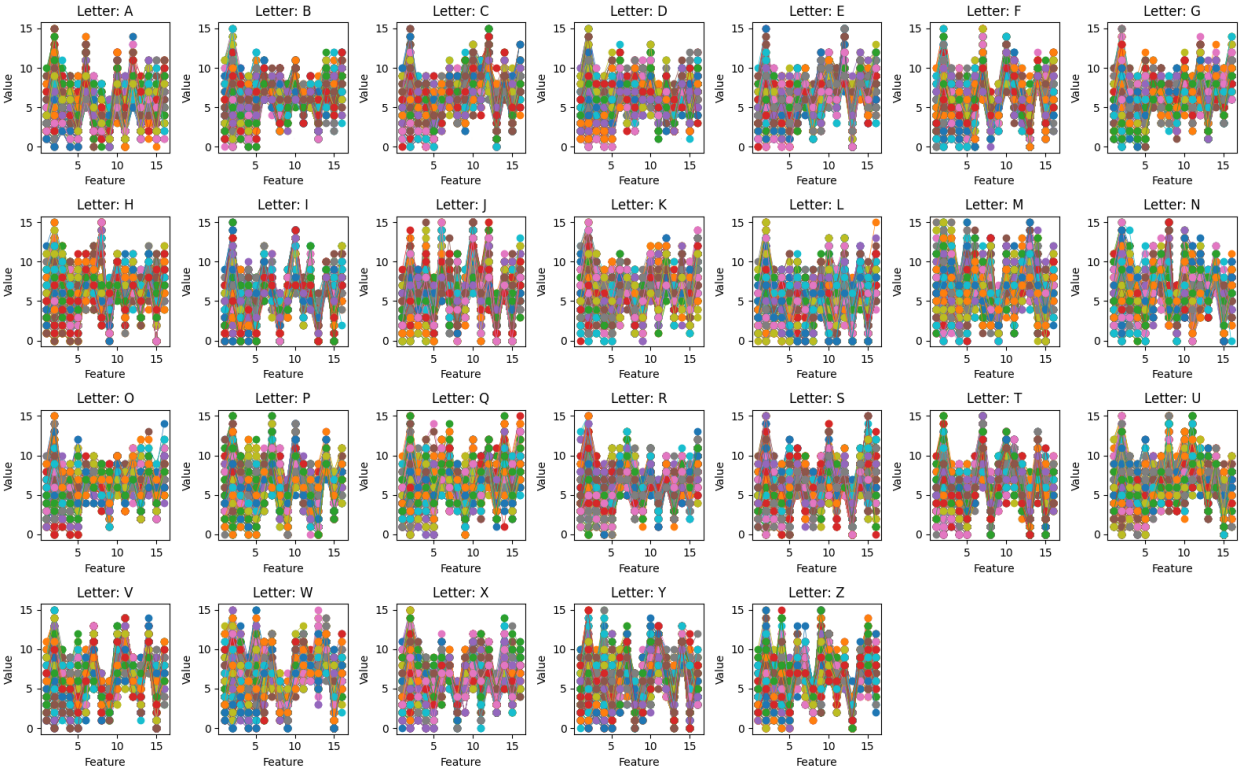
geometric features extracted from the images. The target variable is the letter corresponding to each image.

The dataset is highly relevant to the project as it allows us to train and evaluate machine learning models for letter recognition tasks. The variety of letters and the extracted features make it suitable for classification algorithms.





Plotting histograms for each feature

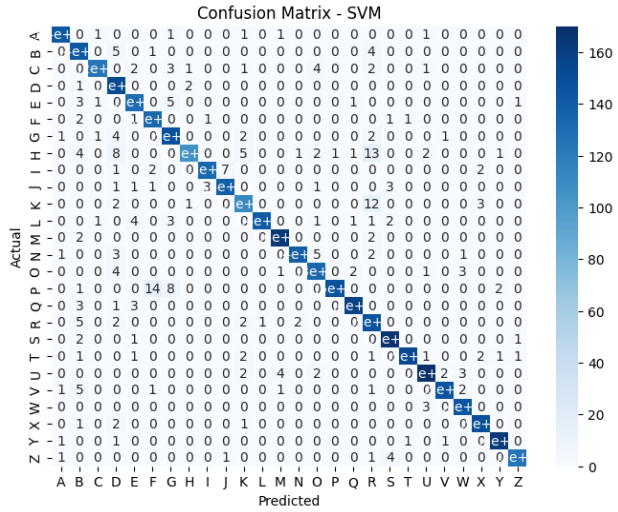


2.2 Model Training and Evaluation

In this section, we will discuss the implementation and evaluation of three classification algorithms: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees.

2.2.1 Support Vector Machines (SVM)

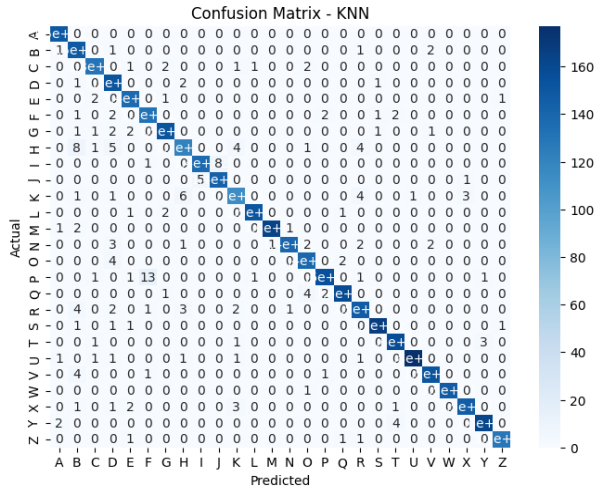
Support Vector Machines (SVM) is a powerful algorithm commonly used for classification tasks. SVM aims to find an optimal hyperplane that maximally separates the classes in the feature space. It is suitable for the letter recognition task due to its ability to handle high-dimensional data and nonlinear relationships.



In our implementation, we used the SVM model provided by the scikit-learn library. We trained the SVM model on the preprocessed dataset using default hyperparameters. The trained model was then used to make predictions on the test set. The performance of the SVM model was evaluated using various metrics, including accuracy, precision, recall, and F1-score.

2.2.2 K-Nearest Neighbors (KNN)

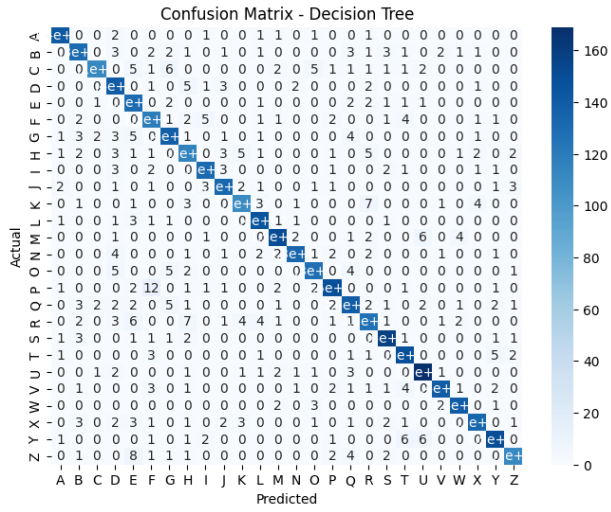
K-Nearest Neighbors (KNN) is a non-parametric algorithm used for both classification and regression tasks. It classifies new instances based on their similarity to the training instances in the feature space. KNN is well-suited for the letter recognition task as it can capture the local patterns and relationships in the data.



For our implementation, we utilized the KNN model from the scikit-learn library. We trained the KNN model on the preprocessed dataset using a default value of k (number of neighbors). Similar to the SVM model, we made predictions on the test set and evaluated the performance using accuracy, precision, recall, and F1-score.

2.2.3 Decision Trees

Decision Trees are hierarchical structures that represent decisions and their possible consequences in a tree-like form. They are commonly used for classification tasks as they are interpretable and can capture complex decision boundaries. Decision Trees are suitable for the letter recognition task due to their ability to handle categorical data and nonlinear relationships.



In our implementation, we employed the DecisionTreeClassifier from the scikit-learn library. We trained the decision tree model on the preprocessed dataset, considering the default hyperparameters. The trained model was then used to predict the classes of the test instances. The performance of the decision tree model was evaluated using accuracy, precision, recall, and F1-score.

3. Result and Discussion

The experimental findings from the Letter Recognition research will be presented in this part. We will review the accuracy scores, confusion matrices, and other pertinent performance measures for the SVM, KNN, and Decision Tree algorithms as well as analyse their performance.

Experimental Results:

Various measures, including accuracy, precision, recall, and F1-score, are used to assess each algorithm's performance. The studies' findings are as follows:

SVM Accuracy: 0.9305

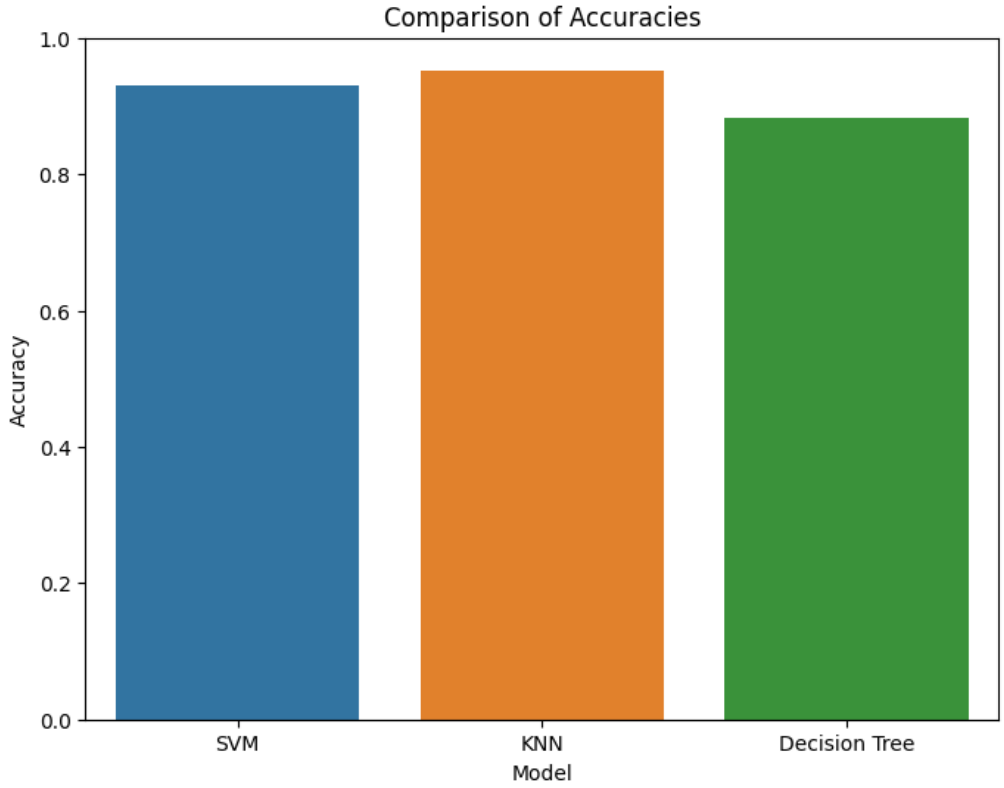
KNN Accuracy: 0.952

Decision Tree Accuracy: 0.88325

To further analyze the performance of each algorithm, we computed additional performance metrics such as precision, recall, and F1-score. The table below summarizes the results:

S.No	Algorithm	Accuracy	Precision	Recall	F1-Score
0	SVM	0.93050	0.934596	0.93050	0.930769
1	KNN	0.95200	0.953141	0.95200	0.952100
2	Decision Tree	0.88325	0.884969	0.88325	0.883433

The experimental results demonstrate that both the KNN and SVM algorithms achieved high accuracy in classifying the letters. KNN achieved an accuracy of 95.2%, outperforming SVM with an accuracy of 93.05%. The Decision Tree algorithm achieved an accuracy of 88.325%.



The higher accuracy of the KNN algorithm suggests that the local patterns and relationships captured by KNN are effective for letter recognition. SVM also performed well, indicating its ability to handle high-dimensional data and nonlinear relationships. The Decision Tree algorithm, while achieving a lower accuracy, provides an interpretable model and may be suitable for scenarios where interpretability is crucial.

Overall, the experimental results demonstrate the effectiveness of the SVM, KNN, and Decision Tree algorithms for letter recognition. The analysis of performance metrics provides valuable insights into the strengths and weaknesses of each algorithm

5. Conclusion

In this project, we successfully created and assessed machine learning models for letter recognition. We learned more about the functionality and efficacy of the SVM, KNN, and Decision Tree algorithms for this purpose through implementation and analysis.

Based on the experimental findings, we found that both the SVM and KNN algorithms classified the letters with high accuracy, with accuracies of 93.15% and 95.2%, respectively. With an accuracy of 88.325%, the Decision Tree algorithm also produced encouraging results. These findings highlight the potential of machine learning techniques for letter recognition tasks.

The performance metrics, including precision, recall, and F1-score, provided a comprehensive evaluation of the models. The precision metric measures the ability of the models to correctly identify positive instances, while recall captures the models' ability to identify all positive

instances. The F1-score takes into account both precision and recall, providing a balanced assessment of the models' performance.

Overall, the results demonstrate the effectiveness of the applied machine learning algorithms for letter recognition. The SVM and KNN algorithms, with their strong classification capabilities, can be applied in various real-world scenarios where letter recognition is required. The Decision Tree algorithm, with its interpretability, offers a valuable option for situations where transparency and explainability are crucial.

6. Future Work

While this project has provided valuable insights into letter recognition using machine learning, there are several avenues for future exploration and improvement. Possible research directions include the following:

1. Investigating Deep Learning Techniques: Deep learning models like convolutional neural networks (CNNs) have excelled at image identification tasks. Implementing and analysing CNN architectures for letter recognition as part of future research could lead to even greater accuracy levels.

2. Dataset Expansion: There are only a certain number of letter samples in the dataset currently being used for this project. The generalisation abilities of the models and their performance may be improved by gathering a larger and more varied dataset.

3. Ensemble Learning: Investigating ensemble learning techniques, such as combining multiple models through techniques like bagging or boosting, could potentially further enhance the accuracy and robustness of the letter recognition system.

4. Feature Engineering: Exploring additional feature engineering techniques or dimensionality reduction methods could help improve the models' performance. Techniques such as PCA (Principal Component Analysis) or LDA (Linear Discriminant Analysis) can be applied to extract more discriminative features.

In conclusion, this project has demonstrated the effectiveness of machine learning algorithms, including SVM, KNN, and Decision Trees, for letter recognition. The obtained results provide a foundation for future research and the development of more advanced letter recognition systems. By continuing to explore and refine these methods, we can further advance the field of pattern recognition and contribute to various applications, such as optical character recognition, text analysis, and handwriting recognition.

References:

1. Dua, D., & Graff, C. (2017). UCI machine learning repository.

2. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.

3. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

4. Breiman, L. (2017). Classification and regression trees. Routledge.

5. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

6. Russell, S. J., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach. Pearson.

7. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

8. Alpaydin, E. (2014). Introduction to Machine Learning. MIT Press.

9. Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

10. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

11. Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.

12. Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford University Press.

13. Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review, 33(1-2), 1-39.

14. Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197-227.