WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Comparative Analysis of Classification Algorithms in Data Mining

[1] Parmar Utsav Nileshbhai, [2] Khokhar Afzal Yusufbhai, [3] Mayank Devani

[1] Computer Engineering, Sal College of Engineering
[2] Computer Engineering, Sal College of Engineering
[3] Assistant Professor Information Technology Department, SAL College of Engineering

ABSTRACT :

As with many data mining projects, classification is one of the primary techniques used in this study. This study aims to compare well-known classification algorithms: Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Naïve Bayer. Evaluation of these algorithms along with their accuracy, precision, recall and F1 score will be calculated for standard datasets. The goal is to figure out which algorithm to use for a specific dataset and its characteristics. Classification is one of the fundamental techniques in data mining, defined as the process of sorting the data into different categories according to the given features. This paper presents a comparative analysis of some of the most known algorithms: Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Naïve Bayes. These algorithms were evaluated using a number of publicly available datasets: Iris, Wine, Credit Card Fraud Detection and Fine Grained Credit Card Fraud Detection using standard metrics like accuracy, precision, recall, F1 score, and confusion matrix analysis. The main goal is to demonstrate the constraints and advantages of each algorithm with respect to dataset characteristics such as size, balance, dimensionality, noise, and prompt practitioners with appropriate model choices.

Introduction

Data mining pertains the ability to extract important information from large databases. Classification is a type of supervised learning which predicts class labels. The classification algorithm selection usually has a profound impact on the results. This paper examines different classification algorithms and analyzes them with respect to different parameters. There is an enormous increase in the amount of data being collected in diverse industries like finance, healthcare, and e-commerce which makes the need for actionable insights important. A portion of knowledge discovery in databases (KDD) is referred to as data mining contributes significantly in changing raw data into useful patterns. One of the ridged supervised learning techniques is classification which is used in data mining to predict discrete variables (class labels).

The performance of classification models relies heavily on the dataset at hand with varying volume of data, features, and even the presence of imbalance among classes. Therefore, it is vital to use an appropriate classification algorithm in order to maximize predictive accuracy and minimize computational cost. The goal of this paper is to analyze and compare the five more widely used classification algorithms in machine learning, discussing their strengths, weaknesses, and practical use across different types of datasets.

Classification Algorithms Overview

Decision Trees

A Decision Tree models choices and the possible results with the aid of a tree-like structure. They are very straightforward to understand and quick for the computer to learn. Each branch of the tree represents a decision or an outcome, whereas the nodes are the results of the decision. CART, ID3, and C4.5 are well-known implementations. The ease of use and low computational need are advantages of Decision Trees but coupled with low interpretability they also pose the weakness of overfitting, specifically with noisy or unbalanced datasets.

Pros:

- Straightforward to comprehend and analyze
- Acceptable for numerical and categorical data
- Minimal data preprocessing is needed

Cons:

- High likelihood of overfitting

- Prone to instability with slight changes in data

Random Forest

Random Forest is an ensemble technique that merges the output of several decision trees to improve accuracy and mitigate overfitting. Random Forest utilizes bagging (bootstrap aggregating) and feature randomness to construct distinct trees, building multiple Decision Trees which are combined for more reliable and accurate forecasts.

Advantages:

- Robustness as well as improved and consistent prediction accuracy
- Overfitting resistance
- Systematic dealing with absent information and disorderly data

Disadvantages:

- Incomparable explanatory power as a Single Decision Tree
- Complex and time-consuming prediction processes against other models

Support Vector Machines (SVM)

SVM looks for the most optimal boundary (hyperplane) to differentiate between various classes. It is efficient in cases where there are many dimensions. SVM finds the optimal hyperplane in a multidimensional space which differentiates different classes. It works well when the number of dimensions is greater than the number of samples. Kernel functions enable non-linear classification with SVM by mapping the inputs into higher dimensional spaces.

Benefits:

- Works in high dimensional feature spaces
- Works effectively when there is a distinct margin of separation.
- Allows the use of various kernels for adaptation.

Drawbacks:

- Can be slow on large datasets because of the number of computations needed.
- Needs to be configured with a specific set of parameters along with the kernel.

K-Nearest Neighbors (KNN)

KNN assigns a data point by the majority label of its k-nearest neighbors. It is easy but computationally costly. KNN is an instance- based learning algorithm that predicts new instances by a majority vote of its k nearest neighbors. It is non-parametric and doesn't need training, but it can be computationally costly at prediction.

Advantages:

- Easy to implement and comprehend
- No training phase (lazy learner)
- Flexible to multiclass classification

Disadvantages:

- Noise and irrelevant features sensitive
- Costly during prediction computationally
- Poor for data of high dimensions (curse of dimensionality)

Naïve Bayes

Naïve Bayes uses Bayes' theorem under the independence assumption of predictors. Naïve Bayes can handle large datasets.
Naïve Bayes is a Bayes' theorem-based probabilistic classifier under the independence assumption of predictors. It is easy and makes a strong assumption about feature independence, yet it tends to work well for text categorization and spam filtering.

Advantages:

- It trains and predicts fast
- It works well on high-dimensional and sparse data
- Works well for text categorization and natural language processing

Disadvantages:

- Assumes feature independence, which rarely holds in practice
- Limited flexibility for modeling complex relationships

Evaluation Metrics

To compare the algorithms, we employ metrics like:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

To compare and measure the classification algorithms objectively, we employ the following metrics:

- Accuracy: Number of correctly predicted instances divided by total instances.
- Precision: Ratio of true positives to predicted positives. Measures model's precision.
- Recall (Sensitivity): Ratio of true positives to actual positives. Measures model's completeness.
- F1-Score: Harmonic mean of precision and recall; balances the trade-off between the two.
- Confusion Matrix: Gives accurate classification results such as false positives and false negatives.

These measures offer a comprehensive measure of model performance beyond accuracy alone, particularly in the case of imbalanced data such as fraud detection.

Results and Comparison

The algorithms were validated using datasets such as Iris, Wine, and Kaggle's Credit Card Fraud Detection. Random Forest and SVM gave good accuracy and stability. Naïve Bayes was fast. KNN gave good performance with balanced data but was weak with large datasets. Experiments were performed using the following datasets:

- Iris Dataset: Low-dimensional, well-balanced, three-class classification problem.
- Wine Dataset: Multivariate input medium-sized dataset, best for multiclass classification.
- Credit Card Fraud Detection: Highly imbalanced binary classification dataset with a large number of features.

Algorithm	Accuracy	Precision	Recall	F1-Score	Notes
Decision Tree	High	Medium	Medium	Medium	Fast training; prone to overfitting
Random Forest	Very High	High	High	High	Robust; best general-purpose option
SVM	High	High	High	High	Excellent for high-dimensional data
KNN	Medium	Medium	Low	Medium	Struggles with large/imbalanced datasets
Naïve Bayes	Medium	Medium	High	Medium	Fast; best for text and large datasets

Insights:

- Random Forest performed excellently on every dataset, particularly in dealing with intricate patterns as well as diminishing overfitting.
- SVM performed strongly with the Iris and Wine datasets, but needed appropriate parameter tuning.
- KNN worked well on balanced, small-sized datasets but experienced scaling problems.
- Naïve Bayes worked very efficiently and effectively on sparse, high- dimensional data such as text.
- Decision Trees gave prompt results but suffered from a lack of robustness compared to ensemble methods.

Conclusion

No single algorithm fits all. The ideal classifier varies with data properties. Random Forest and SVM provide good performance in most scenarios, while Naïve Bayes is suited to text and big data. Deep learning models and ensemble algorithms might be considered in future work for enhancement. This research demonstrates that no classification algorithm performs the best in all cases. A classifier's performance is greatly influenced by dataset properties. Random Forest and SVM provide consistent performance in most general-purpose applications. Naïve Bayes is still a strong contender for large-scale and text-based applications because of its speed and ease of use. KNN and Decision Trees are appropriate for smaller, cleaner data sets where interpretability or low resource consumption is critical.

- Future Work can include:
- Expanding the comparison to deep learning models such as neural networks, CNNs, LSTMs
- Adding feature selection and dimensionality reduction algorithms
- Investigation of the impact of data preprocessing techniques (normalization, SMOTE)
- Investigation of hybrid and stacking ensemble methods

REFERENCES :

1. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques.
2. Scikit-learn Documentation. https://scikit-learn.org/
3. Kaggle Datasets: https://www.kaggle.com/
4. Tan, P.-N., Steinbach, M., & Kumar, V. (2018). Introduction to Data Mining.