NAME: Tejal Mahesh Desai

NJIT UCID: td344

Email Address: [td344@njit.edu](mailto:td344@njit.edu)

24 November, 2024

Professor: Yasser Abduallah

CS 634 101 Data Mining

**Final Term Project Report**

*Implementation and Code Usage*

**Binary Classification using Random Forest Classifier, Convolution Neural Network Classifier and Support Vector Classifier**

**Abstract:**

I investigate several binary classification techniques in this project. In data mining, binary classification is crucial for classifying items that fall into two categories, such as Yes or No or 0 or 1 (in the case of the dataset I used). I evaluate its efficacy and efficiency by putting these algorithms into practice and using a variety of data mining concepts, principles, and techniques. I experiment with different binary classification models (Random Forest Classifier, Convolution Neural Network Classifier, and Support Vector Classifier) using built-in functions, and I learn a lot from the algorithms' accuracy.

**Introduction:**

Data mining is a potent technique for uncovering hidden patterns and relationships within vast datasets. Our project focused on binary classification, a task that involves categorizing data into two distinct groups. This classification is achieved by analyzing the features of each data point and assigning it to one of the two classes.

Algorithms like logistic regression, support vector machines, decision trees, random forests, and neural networks are frequently used for binary classification. To assess the performance of these models, metrics such as accuracy, precision, recall, and F1-score are employed. Binary classification finds applications in various fields, enabling informed decision-making by segregating data into meaningful categories.

In our specific project, we implemented three classification algorithms: Random Forest, 1D Convolutional Neural Network, and Support Vector Machine, to predict loan approval. We evaluated their performance using a confusion matrix and other relevant metrics. To manage computational constraints imposed by the large dataset, we implemented techniques to reduce its size while preserving essential information.

**Core Concepts and Principles:**

**Random Forest Classification**:

Random Forest is a powerful supervised learning technique used for binary classification. It operates by constructing multiple decision trees, each trained on a different subset of the training data. The final prediction is determined by a majority vote among these trees. This ensemble approach significantly reduces overfitting and enhances the model's overall performance.

By leveraging bootstrap aggregation, Random Forest ensures that each tree is built on a diverse sample of the data, leading to a more robust and versatile model. This makes it highly effective for a wide range of classification problems.

In classification tasks, the class selected by the majority of trees becomes the final prediction. For regression tasks, the average prediction of all trees is returned. Random Forest effectively mitigates the overfitting tendency inherent in individual decision trees.

**Convolution 1D**:

Binary classification with 1D Convolutional Neural Networks (CNNs) is a popular approach for data mining tasks involving sequential data, such as natural language processing and time series analysis. This technique excels at identifying local patterns within sequences, making it ideal for tasks like detecting word groupings in text or temporal patterns in time-series data.

The 1D CNN model learns to classify new instances into one of two classes based on the learned patterns. It's crucial to remember that the model's performance is heavily influenced by the quality of the input data, as well as the architecture and hyperparameters of the model itself.

CNNs, in general, are a type of feedforward neural network that employs regularization to automatically learn feature engineering through the optimization of filters or kernels. By using regularized weights and fewer connections, CNNs effectively mitigate the issues of vanishing and exploding gradients that plagued older neural network architectures during backpropagation.

**Support Vector Classifier**:

A Support Vector Machine (SVM) is a supervised machine learning algorithm primarily used for classification tasks, although it can also be applied to regression problems. The core objective of SVM is to identify the optimal hyperplane in a multi-dimensional space that effectively separates data points into different classes. This hyperplane is determined by maximizing the margin, which is the distance between the hyperplane and the closest data points from each class, known as support vectors.

The dimensionality of the hyperplane is directly linked to the number of features in the data. For instance, with two features, the hyperplane is a line, while with three features, it becomes a 2D plane. As the number of features grows, visualizing the hyperplane becomes increasingly challenging.

**Project Workflow:**

My project follows a structured workflow involving various stages and the application of the Apriori Algorithm:

**Data Loading and Preprocessing:**

We begin by loading transaction data from a **Kaggle** dataset, termed <https://archive.ics.uci.edu/dataset/186/wine+quality> .

**Developing Random Forest Classifier:**

For each cross-validation, we split the data and target, fit the model, predict the outcome, calculate the confusion\_matrix, append data into a list for ROC curve, and calculate performance terms.

**Developing Convolution 1D Neural Network Classifier:**

For each cross-validation, we split the data and the target, reshape the data, build and compile the model, compile the model, predict the outcome, calculate confusion matrix, calculate performance terms, append data into the list for ROC curve.

**Developing Support Vector Classifier:**

For each cross-validation, we split the data and target, fit the model, predict the outcome, calculate the confusion\_matrix, append data into a list for ROC curve, and calculate performance terms.

**Crafting the Performance Metrics Formulas without Inbuilt Libraries:**

Having done with the classifiers, following that, I developed the formulas for the performance metrics using the knowledge that I gained from the classroom.

**Displaying the Performance Metrics:**

To display the performance metric tables for all the three algorithms, I used PrettyTable library for command prompt and for ipynb file, I used Ipython.display’s display function.

**Plotting the ROC Curve:**

Lastly, the execution terminates by developing ROC curve for all the three algorithms and saving the developed curves within the plots/ directory (gets created automatically within the cloned folder, if it doesn’t exist).

**Results and Evaluation:**

The project's effectiveness and efficiency are evaluated based on performance measures such as P, N, TP, TN. FP, FN, TPR, TNR, FPR, FNR, Precision, F1, Accuracy, Error Rate, BACC, TSS, HSS, for all the 3 classification algorithms and ROC curves for all within the plots/ directory.

**Conclusion:**

In conclusion, our project demonstrates the application of data mining concepts, principles, and methods. We successfully implemented binary classification. We also calculated performance terms with respect to the three algorithms.

*Prerequisite*

Association Rule Mining code requires [Python](https://www.python.org/) 3.10+ to run (specifically Python 3.10.11).

After having installed Python, follow the commands to execute the code:

A black rectangle with white text

Description automatically generated

*Screenshots*

Images of the Wine Dataset

A screenshot of a computer

Description automatically generated

Importing the libraries

A screen shot of a computer program

Description automatically generated

Utilities class is used to properly format the output in command prompt

A computer screen shot of text

Description automatically generated

Reading the dataset and performing the preprocessing

A screen shot of a computer program

Description automatically generated

Crafting the formulas for Performance metrics

A screen shot of a computer program

Description automatically generated

Developing the Random Forest Classifier

A computer screen shot of text

Description automatically generated

Developing the Support Vector Classifier

A screen shot of a computer program

Description automatically generated

Developing the Convolution Neural Network Classifier

A screen shot of a computer

Description automatically generated

Plotting the ROC Curve

A screen shot of a computer code

Description automatically generated

Displaying the Performance metrics

A screen shot of a computer code

Description automatically generated

Synchronize the execution in main function

A screenshot of a computer program

Description automatically generated

*Execution Screenshots*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a graph

Description automatically generatedA screen shot of a graph

Description automatically generated

*Reasoning the Results*

As per execution, The Random Forest Classifier and Support Vector Classifier (SVC) both achieved the same accuracy of approximately 78.53%, indicating similar effectiveness in classifying the given dataset. Random Forest, an ensemble method, combines multiple decision trees to improve performance and reduce overfitting, making it robust for various types of data. SVC, a kernel-based method, excels in finding the optimal decision boundary, particularly in cases with well-separated classes. On the other hand, the Convolutional Neural Network (CNN) classifier achieved slightly lower accuracy at 75.87%. While CNNs are powerful for image and spatial data, their performance may be less competitive for non-image tabular data unless feature extraction methods align closely with the underlying structure of the data. This highlights the importance of selecting models that suit the nature of the dataset.

*Links and References*

Link to the GitHub repository: <https://github.com/Tejal05131998/Datamining-final.git> (you will be able to access it, once you accept the collaboration request).