ICS3206 - Machine Learning, Expert Systems, and Fuzzy Logic

Course Project

Nathan Bonavia Zammit

Bachelor of Science in

Information Technology (Honours)

(Artificial Intelligence)

January 2023

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I / We*, the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / our* work, except where acknowledged and referenced.

I / We* understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N.B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

	[illegible]
Student Name	Signature
Student Name	Signature
Student Name	Signature
Student Name	Signature
ICS3206	ICS3206 – Course Project
Course Code	Title of work submitted
20/01/2023	
Date	

Introduction

For this assignment, we were given the task of implementing various machine learning algorithms to predict the sex of a speaker from their voice. We were given a close-to-ready-made implementation regarding the following five machine-learning techniques that needed to be implemented [1].

- Artificial neural networks,
- Support vector machines,
- K-Nearest Neighbor,
- Decision trees,
- And logistic regression.

Before implementing said machine learning techniques, the dataset is modified in order to obtain better results. The dataset at hand is called 'voice.csv' which contains various measurements of voices, such as frequency, standard deviation, median, interquartile range, skew, kurtosis, spectral entropy, spectral flatness, mode, centroid, mean fundamental frequency, minimum and maximum fundamental frequency, mean and minimum and maximum duration, and the difference between the maximum and minimum duration, as well as a label indicating whether the voice is male or female.

The code first imports necessary libraries, including pandas for data manipulation, matplotlib and seaborn for data visualization, and scikit-learn for machine learning. Then it loads the data from the 'voice.csv' file and checks for any missing values. It then defines a function to check for unique values in each column and creates a dictionary containing the number of unique values for each column. Next, it separates the data into two sets, 'x' which contains all the measurements, and 'y' which contains the labels. It then plots a pie chart showing the distribution of male and female labels, and finally uses scikit-learn to split the data into a training set and a test set, with the test set being 20% of the total data. This split is done in order to evaluate the performance of a model trained on the training data on unseen data.

Machine Learning Techniques

Decision Tree

A Decision Tree Classifier algorithm from scikit-learn library is used to train a model on the 'x_train' and 'y_train' data (which were obtained by splitting the original dataset into training and test data in the previous code), and then using this trained model to make predictions on the 'x_test' data (which was part of the original dataset that was held out as test data). The model is initialized with a random_state of 1, which is used to set the random seed for the reproducibility of results. The 'fit' function is used to train the model using the training data and then the 'predict' function is used to make predictions on the test data.

The 'score' function is used to evaluate the performance of the model on test data. It returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions from 'sklearn.metrics' are used to generate a report of the precision, recall, f1-score and support for each class, and the overall accuracy respectively. The 'pd.crosstab' function is used to create a confusion matrix.

The parameters of the DecisionTreeClassifier, such as 'random_state', can be changed to see how it affects the performance of the model. For example, increasing the random_state value might increase the accuracy of the model but it might also increase the overfitting.

Random Forest Classifier

A Random Forest Classifier, which is an ensemble method that creates multiple decision trees and combines their predictions to make a final prediction, was implemented in the aforementioned close-to-ready-made implementation [1], and as such was also analyzed and looked into. The algorithm is imported from scikit-learn library and the model is trained on the 'x_train' and 'y_train' data obtained from the previous code. After training, the model is used to make predictions on the 'x_test' data. The model is initialized with a random_state of 200 which is used for reproducibility of results.

The performance of the model is evaluated using the 'score' function, which returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions are used to generate a report of the precision, recall, f1-score and support for each class and the overall accuracy respectively. A confusion matrix is also created using the 'pd.crosstab' function.

Changing the parameters of the RandomForestClassifier, such as increasing the number of trees in the forest, may lead to a better performance of the model but it can also increase overfitting.

Logistic Regression

A Logistic Regression model is a statistical method used for predicting binary outcomes. The algorithm is imported from scikit-learn library and the model is trained on the 'x_train' and 'y_train' data obtained from the previous code. After training, the model is used to make predictions on the 'x_test' data. The model is initialized with a maximum number of iterations of 10000 to ensure convergence, which means that it will run for a maximum of 10000 iterations before stopping.

The performance of the model is then evaluated using the 'score' function, which returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions are used to generate a report of the precision, recall, f1-score and support for each class and the overall accuracy respectively. A confusion matrix is also created using the 'pd.crosstab' function.

It's worth noting that the Logistic Regression algorithm is sensitive to the presence of outliers and non-linear features, thus changing the parameters such as 'max_iter' can affect the model's performance. Increasing the max_iter value might increase the accuracy of the model but it might also increase the overfitting and computation time.

K-Nearest Neighbor

A K-Nearest Neighbors (KNN) algorithm is a non-parametric method used for classification and regression. The algorithm is imported from scikit-learn library and the model is trained on the 'x_train' and 'y_train' data obtained from the previous code. After training, the model is used to make predictions on the 'x_test' data. The model is initialized with the number of nearest neighbors to consider = 15, a distance metric = 'minkowski' and a power parameter = 1. These parameters are used by the algorithm to decide the classification of a new data point.

The performance of the model is then evaluated using the 'score' function, which returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions are used to generate a report of the precision, recall, f1-score and support for each class and the overall accuracy respectively. A confusion matrix is also created using the 'pd.crosstab' function.

It's worth noting that the KNN algorithm is sensitive to the number of nearest neighbors and the distance metric used. Changing these parameters can affect the performance of the model. For example, increasing the number of nearest neighbors might increase the accuracy but it might also increase the computation time. Changing the distance metric can also affect the performance of the model, as different distance metrics can provide different results.

Support Vector Machine

A Support Vector Machine (SVM) model is a supervised learning algorithm that can be used for classification and regression tasks. The algorithm is imported from scikit-learn library and the model is trained on the 'x_train' and 'y_train' data obtained from the previous code. After training, the model is used to make predictions on the 'x_test' data. The model is initialized with a linear kernel which is a simple and efficient choice for linear problems.

The performance of the model is then evaluated using the 'score' function, which returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions are used to generate a report of the precision, recall, f1-score and support for each class and the overall accuracy respectively. A confusion matrix is also created using the 'pd.crosstab' function.

It's worth noting that the SVM algorithm is sensitive to the kernel parameter, which is used to define the decision boundary. Changing the kernel can affect the performance of the model. For example, using a non-linear kernel such as 'rbf' might improve the performance of the model but it might also increase the computation time.

ANN trained by using an MLP

When it came to training an Artificial Neural Network (ANN) as was required, the Multi-Layer Perceptron (MLP) algorithm was chosen. MLP is a type of feedforward artificial neural network that is widely used for supervised learning tasks such as classification and regression. The code was written in a way that can be integrated into the close-to-ready-made implementation without breaking the code. The model is initialized with 3 hidden layers, each having 20 neurons and a maximum iteration of 500. The 'fit' function is used to train the model using the training data and then the 'predict' function is used to make predictions on the test data.

The 'score' function is used to evaluate the performance of the model on test data. It returns the mean accuracy on the given test data and labels. The 'classification_report' and 'accuracy_score' functions from 'sklearn.metrics' are used to generate a report of the precision, recall, f1-score and support for each class, and the overall accuracy respectively. The 'pd.crosstab' function is used to create a confusion matrix.

MLP algorithm is used as it can handle complex patterns and it's also a powerful algorithm that can be used to solve various problems. Changing the parameters such as the number of hidden layers, number of neurons in each layer, and the maximum iteration can affect the performance of the model. For example, increasing the number of hidden layers and neurons in each layer could increase the model's capacity to learn more complex patterns, however, it might also lead to overfitting, and increasing the maximum iteration could increase the accuracy but also the computation time.

Comparison

After training each model, they returned with the following scores:

1. Decision Tree Classifier – Score: 96.52996845425868
2. Random Forest Classifier – Score: 97.79179810725552
3. Logistic Regression Classifier – Score: 90.37854889589906
4. K Nearest Neighbors – Score: 77.4447949526813
5. Support Vector Machine – Score: 92.27129337539432
6. Artificial Neural Network (MLP) – Score: 95.89905362776025

Decision Tree and Random Forest performed relatively well with a score of 96.53% and 97.79% respectively. This is likely due to the fact that both of these models are decision tree-based algorithms, which are known for their ability to handle large amounts of data and make accurate predictions. Random Forest is an ensemble of decision trees which makes it a bit more robust to overfitting and noise in the data, thus it performs a bit better.

Logistic Regression performed relatively poorly with a score of 90.38%. Logistic Regression is a linear model, which means it can only separate the data into two classes by a linear boundary. The problem might be that the data is not linearly separable and thus, the model can't fit the data well.

K-Nearest Neighbors performed poorly with a score of 77.44%. KNN is a non-parametric algorithm, which means it doesn't make assumptions about the underlying distribution of the data. It's sensitive to the noise and the high dimensional data, and it might not perform well if the number of neighbors is not set correctly or if the data is not normalized.

Support Vector Machine performed relatively poorly with a score of 92.27%. SVM is a powerful algorithm that can handle non-linearly separable data by mapping it to a higher dimensional space. However, it's sensitive to the choice of kernel function and the parameters, and it might not perform well if the data is not normalized or if the parameters are not set correctly.

Artificial Neural Network performed relatively well with a score of 95.90%. ANNs are powerful models that can learn complex patterns and make accurate predictions. The MLP algorithm used in this case is a type of feedforward artificial neural network that is widely used for supervised learning tasks such as classification and regression. The model is initialized with 3 hidden layers, each having 20 neurons and a maximum iteration of 500.

It's worth noting that the performance of these models can also depend on other factors such as the quality of the data, the number of samples, the amount of noise in the data, and the choice of the parameters. Additionally, the performance of the models can also be affected by the specific problem and the data set. Finally, when comparing models, it's important to remember that different models might be better suited for different tasks and thus, the best model should be chosen based on the specific problem and requirements.

Statement of Completion

Item	Completed (Yes/No/Partial
Implemented artificial neural network	Yes
Implemented support vector machine	Yes
Implemented k-means clustering	Yes
Implemented decision tree learning	Yes
Implemented logistic regression	Yes
Evaluated artificial neural network	Yes
Evaluated support vector machine	Yes
Evaluated k-means clustering	Yes
Evaluated decision tree learning	Yes
Evaluated logistic regression	Yes
Evaluated logistic regression	Yes

References:

[1] S. Aryaan, "Sex Prediction using 5 Different Models (Beginner)," Kaggle, [Online]. Available: https://www.kaggle.com/code/shabareesharyan/sex-predn-using-5-different-models-beginner/notebook. [Accessed: 20-Jan-2023].