**CA Assignment 1**

**BINARY CLASSIFICATION**

**ASSIGNMENT REPORT SUBMITTED FOR**

**ASSSESMENT OF COMP534**

**(Applied Artificial Intelligence)**

BY

SANTHOSH ARUNAGIRI

201586816

Logo

Description automatically generated

**DEPARTMENT OF COMPUTER SCIENCE**

**UNIVERSITY OF LIVERPOOL**

LIVERPOOL L69 3BX

**INTRODUCTION**

In this assignment, the given dataset is classified using three different classification methods, namely Decision tree, Random Forest, and Support Vector Machine (SVM). We will use Python for data pre-processing and training the models. The dataset consists of 9 feature variables and 2 classes (0 and 1).

**LIBRARIES USED**

We used various Python libraries for data pre-processing, classification, and evaluation.

* “Pandas” is used for data loading, data manipulation and analysis.
* “numpy” is used for mathematical calculations on arrays.
* “matplotlib.pyplot” is used for data visualization.
* “seaborn” is a data visualization library based on matplotlib and provides a high-level interface for creating informative and attractive statistical graphics.
* “sklearn” (scikit-learn) is a machine learning library in Python, which provides tools for data mining and data analysis.
* “train\_test\_split” is a function from sklearn's model\_selection module that splits a dataset into two subsets: training data and testing data.
* “StandardScaler” is a function from sklearn's pre-processing module that scales features to a zero mean and unit variance.
* “SVC” is a support vector machine classifier from sklearn's svm module.
* “RandomForestClassifier” is a random forest classifier from sklearn's ensemble module.
* “DecisionTreeClassifier” is a decision tree classifier from sklearn's tree module.
* “GridSearchCV” is a function from sklearn's model\_selection module that performs hyperparameter tuning using cross-validation.
* precision\_score, recall\_score, accuracy\_score, and f1\_score are functions from sklearn's metrics module used to compute the performance of classification.
* “confusion\_matrix” is a function from sklearn's metrics module that computes a confusion matrix for classification performance evaluation.

The code also sets a warning filter to ignore warnings, which may arise during the execution of the code. And also uses numpy random to provide a seed value, so that the metrics are reproducible(seed 51 is used)

The dataset was split into 80% train data and 20% test data.

**SVM**

For SVM, the 'SVC' class from the 'sklearn.svm' module was used by importing the corresponding library and used GridSearchCV to perform hyperparameter tuning for SVM. created a parameter grid containing various values of 'C', which is the regularization parameter, and 'gamma', which is the kernel coefficient for the 'rbf' kernel. then performed a k=5 kfold cross-validation on the train data to evaluate the performance of each combination of hyperparameters and selected the combination that gave the best performance. Which is then tested on the 20% data and the performance of the classification is noted with respect to accuracy, precision, recall, F1 score, and the confusion matrix is plotted.

**RANDOM FOREST METHOD**

For Random Forest, the 'Randomforestclassifier' class from the 'sklearn.ensemble' module was used by importing the corresponding library and used GridSearchCV to perform hyperparameter tuning. created a parameter grid containing various values of 'n\_estimators', which is the number of trees in the forest, and 'max\_depth', which is the maximum depth of the trees then performed a k=5 kfold cross-validation on the train data to evaluate the performance of each combination of hyperparameters and selected the combination that gave the best performance. Which is then tested on the 20% data and the performance of the classification is noted with respect to accuracy, precision, recall, F1 score, and the confusion matrix is plotted.

**DECISION TREE**

For Decision Tree, the 'DecisionTreeClassifier' class from the 'sklearn.tree' module was used by importing the corresponding library. Hyperparameters were tuned using GridSearchCV. A parameter grid containing various values of 'max\_depth' was created, which is the maximum depth of the decision tree. A k=5 k-fold cross-validation was performed on the train data to evaluate the performance of each combination of hyperparameters, and the combination that gave the best performance was selected. The selected model was then used to make predictions on the 20% test data, and the performance of the classification was evaluated using accuracy, precision, recall, F1 score, and the confusion matrix was plotted to visualize the results.

**CONFUSION MATRIX**

* METHOD 1

The confusion matrix for Random Forest classification is a 2x2 matrix, where the rows represent the actual labels, and the columns represent the predicted labels. The matrix is as follows:

|  |  |  |
| --- | --- | --- |
|  | **PREDICTED NEGATIVE** | **PREDICTED POSITIVE** |
| **ACTUAL NEGATIVE** | 81 | 5 |
| **ACTUAL POSITIVE** | 3 | 51 |

In this matrix, the true negatives (TN) are 81, which means that 81 samples were correctly classified as negative. The false positives (FP) are 5, which means that 5 samples were wrongly classified as positive when they were actually negative. The false negatives (FN) are 3, which means that 3 samples were wrongly classified as negative when they were actually positive. Finally, the true positives (TP) are 51, which means that 51 samples were correctly classified as positive.

* METHOD 2

The confusion matrix for SVM classification is a 2x2 matrix, where the rows represent the actual labels, and the columns represent the predicted labels. The matrix is as follows:

|  |  |  |
| --- | --- | --- |
|  | **PREDICTED NEGATIVE** | **PREDICTED POSITIVE** |
| **ACTUAL NEGATIVE** | 82 | 4 |
| **ACTUAL POSITIVE** | 3 | 51 |

In this matrix, the true negatives (TN) are 82, which means that 82 samples were correctly classified as negative. The false positives (FP) are 4, which means that 4 samples were wrongly classified as positive when they were actually negative. The false negatives (FN) are 3, which means that 3 samples were wrongly classified as negative when they were actually positive. Finally, the true positives (TP) are 51, which means that 51 samples were correctly classified as positive.

* METHOD 3

The confusion matrix for Decision Tree classification is a 2x2 matrix, where the rows represent the actual labels, and the columns represent the predicted labels. The matrix is as follows:

|  |  |  |
| --- | --- | --- |
|  | **PREDICTED NEGATIVE** | **PREDICTED POSITIVE** |
| **ACTUAL NEGATIVE** | 81 | 5 |
| **ACTUAL POSITIVE** | 4 | 50 |

In this matrix, the true negatives (TN) are 81, which means that 81 samples were correctly classified as negative. The false positives (FP) are 5, which means that 5 samples were wrongly classified as positive when they were actually negative. The false negatives (FN) are 4, which means that 4 samples were wrongly classified as negative when they were actually positive. Finally, the true positives (TP) are 50, which means that 50 samples were correctly classified as positive.

**METRICS OF ALL CLASSIFIERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLASSIFIERS** | **ACCURACY** | **F1 SCORE** | **PRECISION** | **RECALL** |
| **RANDOM FOREST** | 0.942857 | 0.943040 | 0.943622 | 0.942857 |
| **SVM** | 0.950000 | 0.950083 | 0.950267 | 0.950000 |
| **DECISION TREE** | 0.935714 | 0.935821 | 0.936028 | 0.935714 |

The best classifier based on accuracy is : SVM.

The best classifier based on F1 score is : SVM.

The best classifier based on precision score is : SVM.

The best classifier based on recall score is : SVM.

Overall, we can conclude that SVM and Random Forest perform similarly well on this task, with SVM having slightly higher accuracy and Precision, F1 Score and recall. The Decision Tree performs slightly worse than the other two classifiers on this task.

The reason for this is that it depends on a number of variables, complexity of the data and the amount of features, some classification methods perform better than others in a particular dataset. it appears that SVM and Random Forest perform better than Decision Tree in this dataset.

Usually, Decision trees, can be sensitive to the specific structure of the data and may require additional tuning to perform well. Decision Trees are simple and easy to method to classify. They work well when the relationship between features is relatively simple and linear. However, they can easily overfit the data and may require reducing or other regularization techniques to prevent this.

Regarding my reflection on this project, I think it is a great training to evaluate and compare different classification methods on a specific task. Through this project, I learned how to use different performance metrics such as Accuracy, F1 Score, Precision, and Recall to evaluate the performance of classifiers. using different libraries, learning how the module in library works and plotting graphs Additionally, I learned the importance of data pre-processing, feature engineering, and hyperparameter tuning in improving the performance of classifiers. It was very informative to learn the usage of k fold cross validation.

**REFERENCES**

1. Waskom, M. (2021). seaborn: statistical data visualization — seaborn 0.10.1 documentation. [online] seaborn.pydata.org. Available at: https://seaborn.pydata.org/index.html.
2. scikit-learn.org. (n.d.). About us — scikit-learn 0.22.2 documentation. [online] Available at: https://scikit-learn.org/stable/about.html [Accessed 29 Apr. 2020].
3. Scikit-learn.org. (2019). sklearn.model\_selection.GridSearchCV — scikit-learn 0.22 documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html.
4. scikit learn (2018). 1.4. Support Vector Machines — scikit-learn 0.20.3 documentation. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/svm.html.
5. Scikit-learn (2018). 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.
6. scikit learn (2019). sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation. [online] Scikit-learn.org. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>.
7. scikit-learn (2019). scikit-learn: machine learning in Python. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/.
8. NumPy (2022). Overview — NumPy v1.19 Manual. [online] numpy.org. Available at: <https://numpy.org/doc/stable/>.
9. pandas.pydata.org. (n.d.). pandas documentation — pandas 1.0.1 documentation. [online] Available at: https://pandas.pydata.org/docs/ [Accessed 27 Feb. 2020].
10. matplotlib.org. (n.d.). Overview — Matplotlib 3.4.1 documentation. [online] Available at: https://matplotlib.org/stable/contents.html [Accessed 18 Apr. 2021].
11. seaborn (2012). seaborn: statistical data visualization — seaborn 0.9.0 documentation. [online] Pydata.org. Available at: https://seaborn.pydata.org/.
12. Towards Data Science. (2019). Towards Data Science. [online] Available at: <https://towardsdatascience.com/>.
13. Scikit-learn.org. (2019). sklearn.model\_selection.GridSearchCV — scikit-learn 0.22 documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>.
14. scikit-learn developers (2019). sklearn.svm.SVC — scikit-learn 0.22 documentation. [online] Scikit-learn.org. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.
15. Scikit-learn (2018). 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.