**Analysis of Dataset using Machine Learning Algorithms**

**Introduction:**

This report presents a detailed analysis of the dataset using various machine learning algorithms. The main objective is to evaluate the performance of logistic regression, decision tree, and random forest classifiers on the dataset. Analysis includes data exploration, visualization, preprocessing, model training, hyperparameter tuning, and performance.

**Libraries Imported:**

The imported libraries play a crucial role in analyzing and modeling the dataset. The Pandas library and the NumPy library are used to manipulate the data and perform numerical operations, which serve as the basis for the data exploration and the preprocessing of the data. The Matplotlib library and the Seaborn library make it possible to visualize the data’s distribution and the relationships between the variables through different plots. Scikit-learn provides machine learning algorithms such as decision tree classification, random forest classification, logistic regression and data splitting tools. It also provides tools for hyperparameter fine tuning, model evaluation, and more. Imblearn’s SMOTE module deals with class imbalance and sklearn’s metrics provide functions to evaluate the model’s performance, including accuracy score, class report, confusion matrix, etc. Lastly the Warnings module was imported to supress warnings for cleaner output. Together, all of these libraries form a powerful framework for analyzing and modeling the data.

**Data Exploration:**

The dataset, named **'dataset\_assignment1.csv'**, was loaded using the pandas library. Initial exploration revealed key insights such as the dataset dimensions **(700 rows and 10 columns)**, column names, summary statistics such as the count, mean, standard deviation, minimum value, 25th percentile (first quartile), median (50th percentile), 75th percentile (third quartile), and maximum value. The missing values, unique values per column, and class distribution are other things evaluated for the preprocessing. On exploration we find the last column named class is the target class. **data[‘class’].value\_counts()** provides me with the information about the number of samples that belong to each class that is **459 samples for class 0 and 241 samples for class 1**. We find that the dataset is imbalanced and we resolve this issue using the **SMOTE** technique later during the preprocessing of the dataset. This step provided a foundational understanding of the dataset's characteristics.

**Data Visualization:**

To visualize the class distribution of the dataset, seaborn count plots and matplotlib pie charts were created. For each feature, histograms were created to understand the distribution of the class and a boxplot was created to identify the outliers. Boxplots show the median, the quartiles, and the outliers in a dataset. They allow to quickly assess the central tendency, the spread, and the skewness of a dataset. Because boxplots show the range and the concentration of the data points, they make it easy to compare groups and help in the outlier detection. Using these visualizations, I was able to gain valuable insights into the data structure of the dataset. It also helped to select features and preprocess the data.

**Data Preprocessing:**

Preprocessing steps include: Distinguishing the features and target variable, distributing the dataset into training and testing set **(80:20 ratio)** with the **train\_train\_split** function, addressing class imbalance with **SMOTE Synthetic minority over-sampling** technique, generating synthetic samples for minority class ensuring balanced representation in training data. Applying SMOTE after splitting the dataset into training and testing sets prevents data leakage and ensures unbiased evaluation of model performance by generating synthetic samples only from the training data. To show that the data is now balanced and better trained for classification purpose, I have plotted another pie chart depicting the class distribution.

**Model Training and Evaluation:**

The dataset that underwent preprocessing was utilized to train three distinct machine learning algorithms: Logistic Regression, Decision Tree, and Random Forest classifiers. Hyperparameter tuning and identification of the best parameters for each model were carried out using **RandomizedSearchCV**. Each of the machine learning algorithms have undergone hyperparameter tuning and the results have been compared.

**Logistic Regression** uses the **'liblinear'** solver algorithm for optimization, sets the random seed to 42 for better reproducibility, applies **'L2'** regularization for stable and generalised model fitting, limits optimization iterations to 100, fits an intercept term accounting for potential shifts in the decision boundary, and treats all classes equally by setting class weights to None. The accuracy score stays the same.

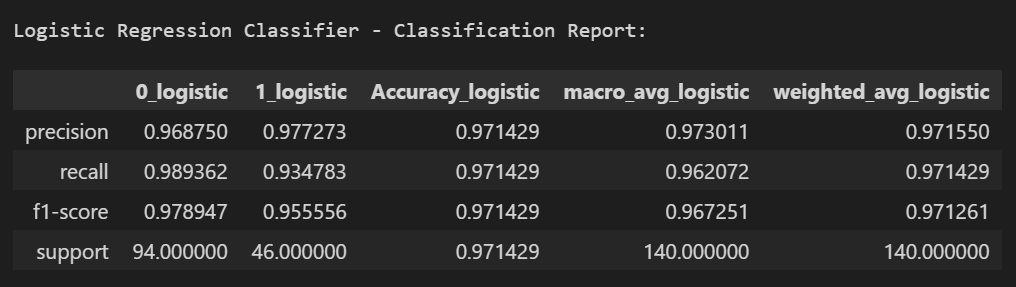
**Decision Trees** make use of the **'random'** splitter approach, set the minimum number of samples required to split a node at 4, limit the tree's maximum depth to 5 levels, and employ the **'entropy'** criterion to evaluate the effectiveness of a split.

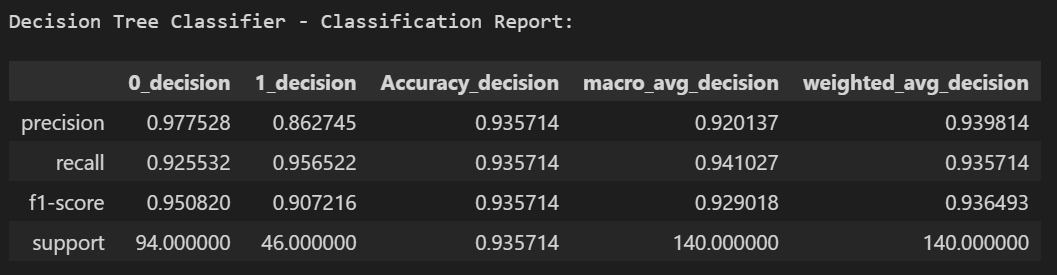
**Random Forest** utilizes **200 decision trees** for ensemble learning, establishes the minimum number of samples required to split an internal node at 2, restricts the maximum depth of the trees to 5 levels, and utilizes the **'gini'** criterion for impurity measure. The accuracy score displayed significant improvement.

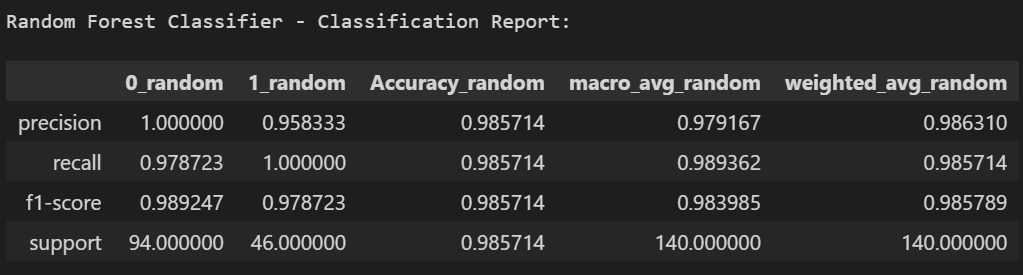
The trained models were evaluated based on their accuracy scores, classification reports, and confusion matrices. These metrics provided insights into the models' performance and their ability to generalize to unseen data.

**Results and Discussion:**

The logistic regression classifier achieved an accuracy of 0.9714, while the improved logistic regression classifier also obtained the same accuracy of 0.9714, which remains consistent. On the other hand, the decision tree classifier yielded an accuracy of 0.9214, whereas the enhanced decision tree classifier achieved a slightly higher accuracy score of 0.9357. Similarly, the random forest classifier attained an accuracy of 0.9714, and the enhanced random forest classifier further improved it to 0.9857. The classification reports provided a comprehensive analysis of the models' performance, including precision, recall, F1-score, and support for each class. The confusion matrices visually depicted the models' predictions, highlighting their strengths and weaknesses. Below are the classification reports for each of the classification algorithms:

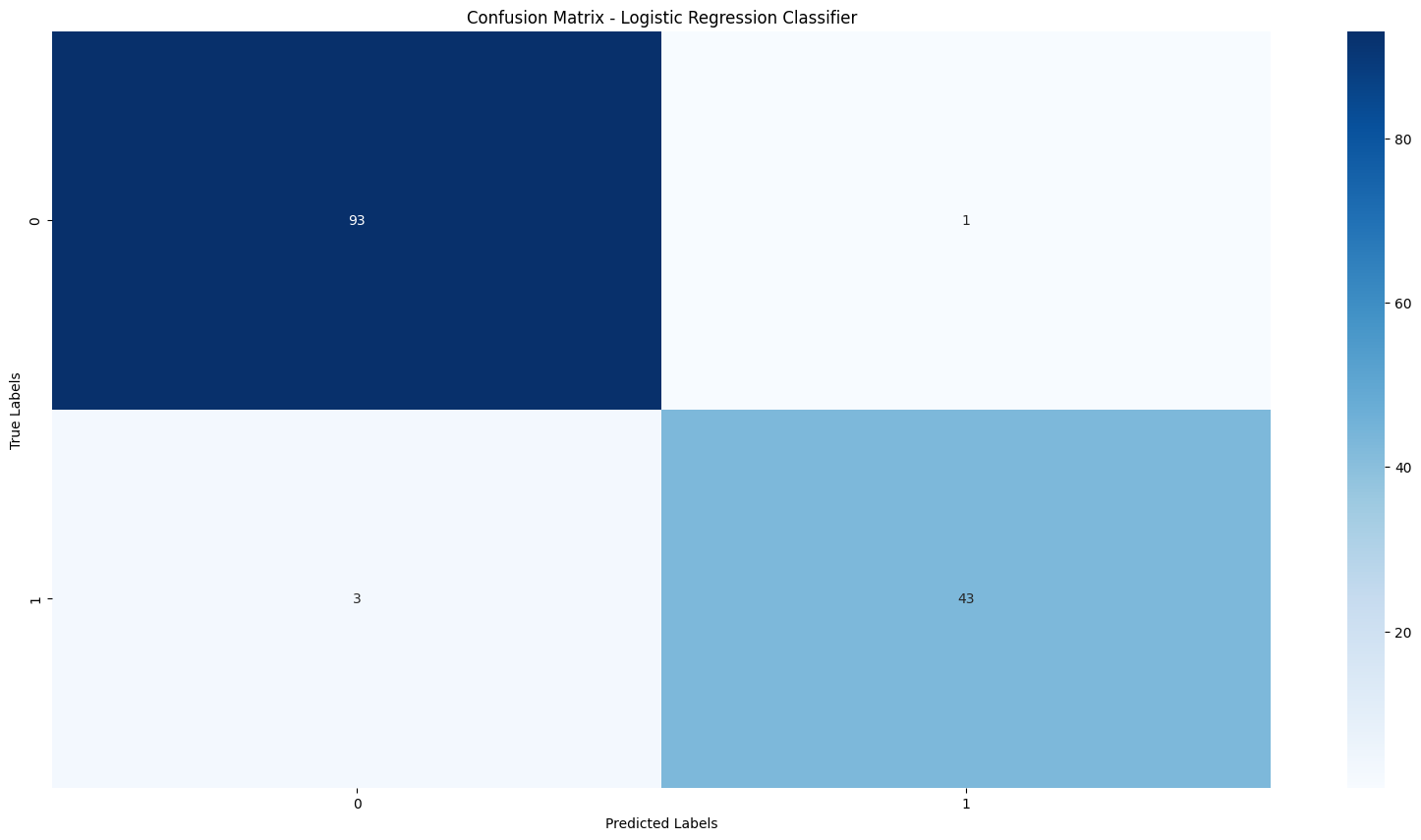




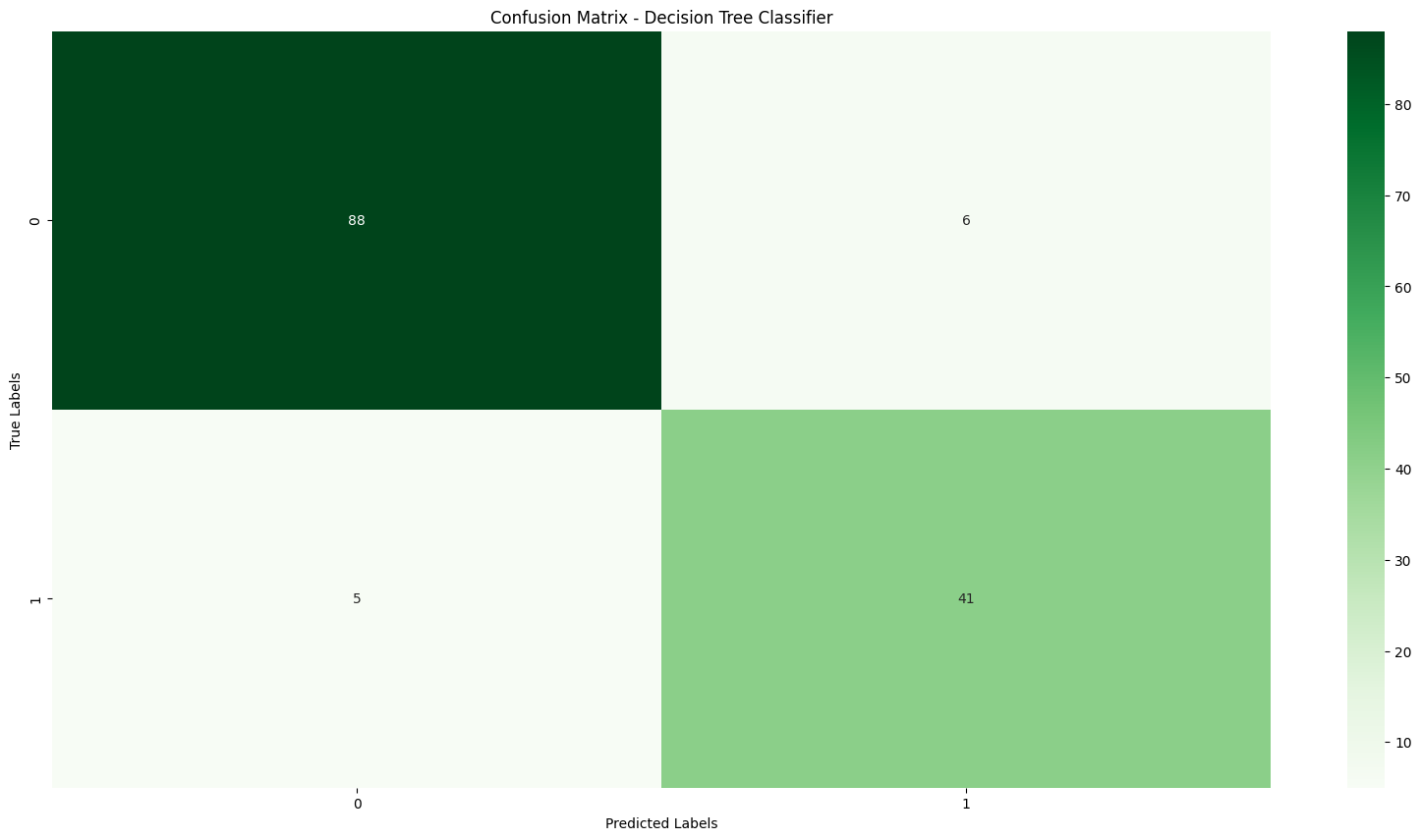


Each classifier's report contains various performance metrics like precision, recall, f1-score, and support for two classes (class 0 and class 1), in addition to overall accuracy and macro and weighted average scores for each metric. In Logistic Regression, class 0 shows greater precision, whereas class 1 displays higher recall, indicating a trade-off between the two metrics. Conversely, the Decision Tree classifier demonstrates a harmonious blend of precision and recall for both classes, albeit with slightly lower values than the Logistic Regression. Meanwhile, the Random Forest classifier demonstrates perfect precision for class 0 and impressive scores across all metrics, suggesting it could potentially be the most effective model among the three based on these findings. Each metric plays a vital role in assessing model performance, with support indicating the number of true instances for each class, and the averages offering a comprehensive overview of the overall performance, considering the balance between classes.

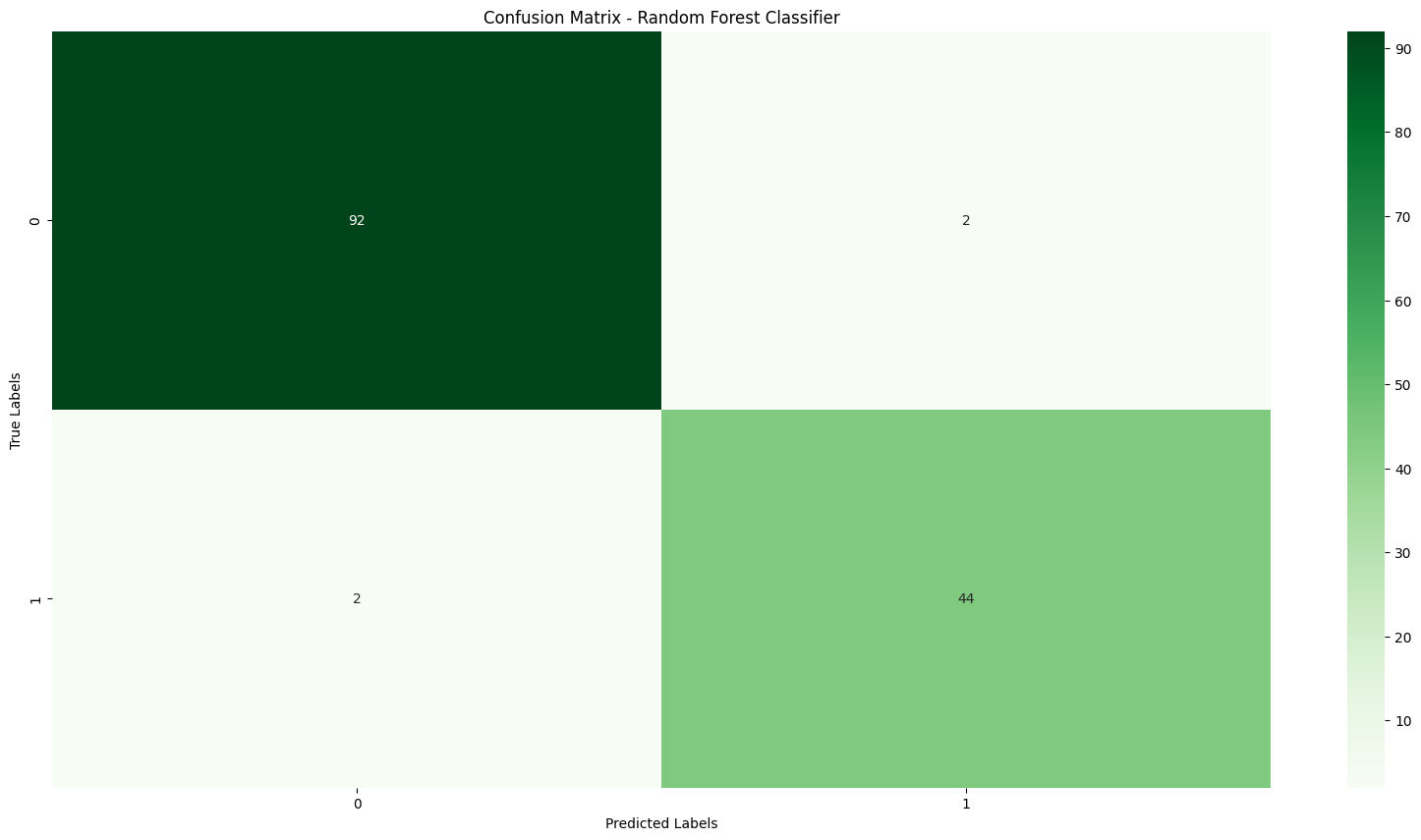
**Confusion Matrices Explanation:**

****

The matrix of the Logistic Regression model shows 93 correct predictions for class 0 and 43 correct predictions for class 1, along with 1 incorrect prediction for class 0 and 3 incorrect predictions for class 1.

****

The Decision Tree model displays 88 true positives for class 0 and 41 true negatives for class 1 in the matrix, as well as 6 false negatives for class 0 and 5 false positives for class 1.



The confusion matrix in the Random Forest model reveals that it accurately predicted 92 instances of class 0 and 44 instances of class 1 (true positives and true negatives). However, it incorrectly classified 2 instances of class 0 as class 1 (false positives) and 2 instances of class 1 as class 0 (false negatives).

Confusion matrices are employed to visually illustrate the performance of classification models. The main diagonal displays the number of accurate predictions made by the model for each class, while the remaining cells indicate where the model made inaccurate predictions. A higher count on the main diagonal compared to the off-diagonal numbers suggests superior performance. Each model demonstrates varying levels of performance, with the Random Forest classifier exhibiting a significant number of true positives and true negatives, indicating reliable performance. The Logistic Regression model excels in correct predictions for class 0 and displays strong overall performance. On the other hand, the Decision Tree model exhibits more misclassifications than the other two models, suggesting a potential requirement for further tuning or increased model complexity.

**Conclusion:**

While the Random Forest Classifier demonstrates superiority in terms of the metrics presented (achieving the highest accuracy of 98.57%, along with perfect precision and recall for the "1" class), it is essential to recognize that the selection of a classifier heavily depends on the specific application and dataset. Factors such as interpretability and computational cost also play a significant role. Further analysis reveals that Random Forests excel in handling complex datasets and preventing overfitting due to their ensemble nature, making them proficient in both classification and regression tasks. On the other hand, Support Vector Machines (SVMs) perform well in high-dimensional spaces with smaller datasets, while Logistic Regression provides a simpler and more interpretable model. In this particular case, the Random Forest may have leveraged the characteristics of the dataset, potentially having sufficient data points to mitigate its tendency to overfit. This assignment emphasizes the importance of evaluating different classification algorithms to identify the most suitable one.

Through this process, I have acquired valuable insights:

**Learnings:** I have developed a deeper understanding of the advantages and disadvantages associated with various classification methods, as well as the crucial role of performance metrics in selecting the optimal model.

**Areas for Improvement:**

**Data Pre-processing:** A more thorough exploration of data cleaning and feature engineering techniques could potentially enhance model performance.

**Hyperparameter Tuning:** Extensive optimization of hyperparameters for each model could lead to further enhancements.

**Alternative Algorithms**: Experimenting with alternative classification methods, such as Gradient Boosting Machines, may offer additional valuable insights. By integrating these considerations into future projects, I aim to create more resilient and widely applicable models.