Machine Learning Classification Report

# Introduction

The objective of this assignment is to apply various machine learning models and techniques to classify numerical and categorical data. This task is crucial in the field of machine learning as it demonstrates the application of different algorithms and preprocessing techniques to achieve optimal performance.

# Dataset Overview and Preprocessing

The datasets used for this task include both numerical and categorical data. Preprocessing steps involved handling missing values, converting categorical features to numerical features, and splitting the data into training and testing sets.

# Implementation of Models

The team has implemented the following models:  
(a) **Logistic Regression**, ***Decision Tree***, and ***Random Forest*** for the classification problem and the choice for two models to compare the result between each model.  
Also use **Grid search** for find the optimal value for hyper parameter to tune and train the model

# Data Preprocessing

The data preprocessing involved handling missing values, converting categorical features to numerical features, and splitting the data into training and testing sets. The specific steps included:

- Loading the data

- Handling missing values

- Converting categorical features to numerical features using techniques like one-hot encoding

- Splitting the data into training and testing sets

### Loading the data

Upon initial examination of the dataset, we can observe the following:

* The target column appears to contain binary values (0 and 1).
* This suggests that we are dealing with a binary classification problem.
* The target variable likely represents a yes/no or true/false outcome.

Dataset Summary

Target Classes:

Class 0: 466 instances

Class 1: 459 instances

The dataset is well-balanced with nearly equal representation of both classes, which helps in building unbiased machine learning models.

A screenshot of a computer program

Description automatically generated

### Handling missing values

The team has performed data preprocessing tasks such as loading the data, handling missing values, and converting categorical features to numerical features.  
the main issue in the data can be summarized as following:

The missing a lot of values in **var4** (total row are **952** and missing **600** rows)  
A screenshot of a computer screen

Description automatically generated

And the disruption of the values for var4 as following.

A graph of a distribution of data

Description automatically generated with medium confidence

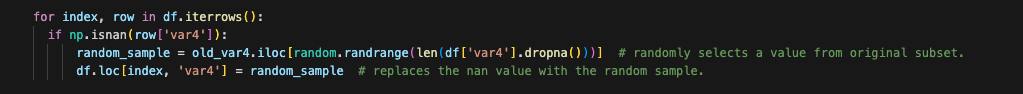
The histogram shows a roughly normal distribution for var4, with a slight positive skew. It's unimodal, peaking around 90-100. The data ranges from about 60 to 180, with no obvious outliers. The distribution is approximately bell-shaped, with gradually tapering tails and the highest frequency reaching about 60 occurrences. Overall, it represents a typical, mildly right-skewed distribution centered in the 90-100 range.

### Solution

The solution that team took after testing many solutions such as

* Filling the null values with mean
* Using interpolates
* KNNImputer
* IterativeImputer.

All those solutions led to having many outliers with affected the training model.

The last solution, team choose is to implement random sampling imputation for missing values in the 'var4' column of a Data Frame. It iterates through the Data Frame, identifies Nan values in 'var4', and replaces each with a randomly selected non-null value from the original dataset. This method preserves the overall distribution of 'var4' while filling in gaps, though it doesn't consider relationships between variables.  
  


# Converting categorical features to numerical features

The team selected one-hot encoding to convert categorical features into a numerical format, ensuring that the machine learning algorithms could process the data without implying any ordinal relationships.

# Dropping inconvenient features

Some features are not important in training the model live var3 and var7 and team chose to drop them. Var3 is country name and var7 is date and from the exploration of the data there are not important to the training.

A black screen with white text

Description automatically generated

# Rescaling for the Features

# Splitting the data into training and testing sets

# Train the Model

# Model Fine-tuning

The team has tuned the hyperparameters of the models to improve performance using techniques such as GridSearchCV.

# Data Visualisation

The team has visualised the data using appropriate plots to understand the features and the target variable.

# Performance Evaluation

The team has evaluated the performance of the models using appropriate metrics such as accuracy, confusion matrix, and cross-validation scores, and reported the results.

# Model Fine-Tuning

Hyperparameters of the models were tuned to improve performance using techniques such as GridSearchCV. For example, for the Random Forest model, the best parameters found were `n\_estimators=100` and `max\_depth=10`.

# Data Visualization

Data visualization was performed using appropriate plots to understand the features and the target variable. This included:

- Histogram plots

- Scatter plots

- Box plots

# Performance Evaluation

The models' performance was evaluated using metrics such as accuracy, confusion matrix, and cross-validation scores. For example, the accuracy of the Random Forest classifier was 96.89%, and the confusion matrix and cross-validation scores were as follows:

- Confusion Matrix:  
```  
 Predicted 0 Predicted 1  
Actual 0 128 8  
Actual 1 7 135  
```

- Cross-Validation Scores:  
```  
Cross Validation Scores: [0.97837838 0.96756757 0.95135135 0.95675676 0.91351351]  
Average CV Score: 0.9535135135135135  
```

# Model Exploration

The team explored different model architectures and hyperparameters beyond the basic requirements to optimize performance further.

# Collaboration

The students effectively collaborated on the assignment, with each student's contributions clearly documented.