Track 2: Online Shopper Conversion Prediction

Given a person's website activity and historical analytics, can we predict whether they will make a purchase?

Dataset

The dataset can be accessed here. Refer `metadata.txt` for details about the data fields.

Non-Competitive Part (9 Marks)

2. Exploratory Data Analysis (EDA)

1. **Understand the dataset:** Load the dataset and display its basic structures (`.head()`, `.info()`, `.describe()`).
2. **Visualise the distribution of features:** Use histograms for numerical variables and count plots for categorical features.
3. **Compare the distribution of features for each target class:** Use violin plots for numerical variables and stacked bar plots for categorical variables.

2. Data Preprocessing

1. **Remove outliers:** Use boxplots and the interquartile range (IQR) method to identify and remove outliers.
2. **One-hot encoding:** Convert categorical variables to numerical ones using one-hot encoding.
3. **Standardize numerical variables:** Apply z-score normalization using `StandardScaler` from `sklearn`.

3. Logistic Regression

1. **Data splitting:** Split the data into training and testing sets (80:20 split) without randomization.
2. **Implement Logistic Regression from scratch:** Write the algorithm from first principles and train the model.
3. **Performance evaluation:** Report accuracy, precision, recall, and F1 scores on the test set.
4. **Comparison:** Compare results with the default `sklearn` implementations of `LogisticRegression`, `SVC` and `DecisionTreeClassifier`.

Competitive Part (6 Marks)

We will be using **F1-score** as the metric to determine rankings.

1. **Feature Engineering and Selection:** Analyse correlations between features and drop highly correlated or low-variance features. Engineer new features as needed.
2. **Handle class imbalance:** Explore techniques such as oversampling (`SMOTE`), undersampling, or class-weight adjustment to address class imbalance.
3. **Experiment with advanced algorithms:** Use variations of `Decision Trees`, `SVMs`, `Bagging`, and `Boosting` to improve performance. Implement these techniques using `sklearn` and optimize hyperparameters to maximize F1-score.

NOTE: In Competitive Part, you are free to **play with the data**. The **type of models** should be restricted to as stated.