Name – Dhirendra Kumar

Roll no- 24AR10024

# Logistic Regression and Random Forest Classifier

## 1.Approach

The task was to build and evaluate two classification model on the given small dataset (1800 rows). here is how I did it.

1. **Data preprocessing**: checked the missing values, duplicate rows, performed operations df.info(), df.describe(), df.corr(), separated features into numerical and categorical variables.
2. **EDA**
3. Univariate Analysis:

First, analysis on target variable was performed to check distribution of classes and found the variable imbalanced with 70.8% as class 0 and 29.2% as class1.

Then plotted countplot for categorical variables.

For numerical variables I plotted histplot so check their distribution and found that two features Cart\_value and Browser\_Refresh\_Rate is right skewed.

The boxplots for numerical variables revealed the presence of outliers which can affect the performance of Logistic Regression model.

1. Multivariate Analysis:

After plotting the pair-plots strong & positive correlation was found. Which can be confirmed from the correlation heatmap below it. From the heatmap it can also be concluded that the features in data didn’t have very high correlation with the target variable, with almost all Corr<0.4.

1. **Feature Engineering**

* The features Browser\_Refresh\_Rate, Cart\_value, Pages\_viewed were clipped to remove the outliers present in data.
* Log transformation (np.log1p) was applied on right skewed variables Pages\_viewed and Cart\_value .
* Various new features was created from existing.
* At last one hot encoding and scaling of features was done.

1. Model selection: Logistic Regression and Random Forest was chosen.
2. Hyperparameter tuning: RandomizedSearchCV was used for both models.
3. Evaluation: Accuracy, f1\_score and confusion matrix.

## Model explanations

1. **Logistic Regression**

* it is a linear model used for binary classification.
* It predicts probability of belonging to positive class by using sigmoid function.
* Weights and biases are updated using gradient descent algorithm by minimizing log loss function.

1. **Random Forest**

* It is an ensemble of decision trees trained on bootstrap samples.
* Each tree does splitting on random subset of features.
* Loss function for each node is either gini impurity or Entropy loss.
* Final prediction is done by voting and the class with majority votes is chosen.

## Key Insights and Performance Analysis

* Logistic regression is benefitted from standardization and clipping of outliers.
* Random forest initially overfit the data but after tuning its generalization improved.
* Features with low correlation helped the logistic regression and random forest captured the non-linear relation of data.

## Model Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Train Accuracy | Test Accuracy | F1 score |
| Logistic regression(before) | 0.733889 | 0.711111 | 0.630682 |
| Logistic regression(after) | 0.740556 | 0.762222 | 0.658147 |
| Random forest(before) | 1.000000 | 0.800000 | 0.648438 |
| Random forest(after) | 0.772222 | 0.746667 | 0.664706 |

* Clearly the logistic regression model improved after hyperparameter tuning
* Initially the random forest model was overfitting but after tuning it generalized better.
* However, the test accuracy of Random forest model before tuning was best of all.

## Conclusion

The performance of model plateaued likely due to:

* Small size of dataset (1800 rows)
* Feature saturation
* Accuracy ceiling reached