MSC 571 Business Analytics & AI
Midterm Write-up
Malicious Web Pages Case Study

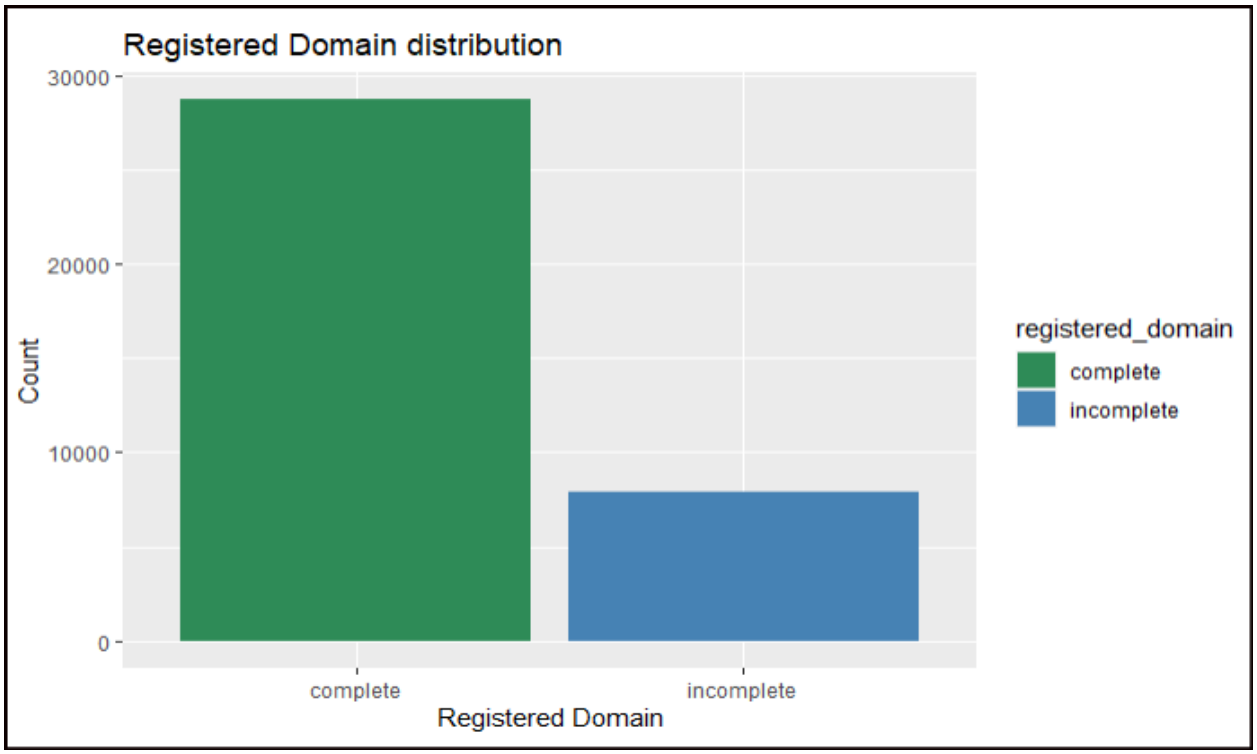
Manju Penumarthi
#A25319647

OBJECTIVE:

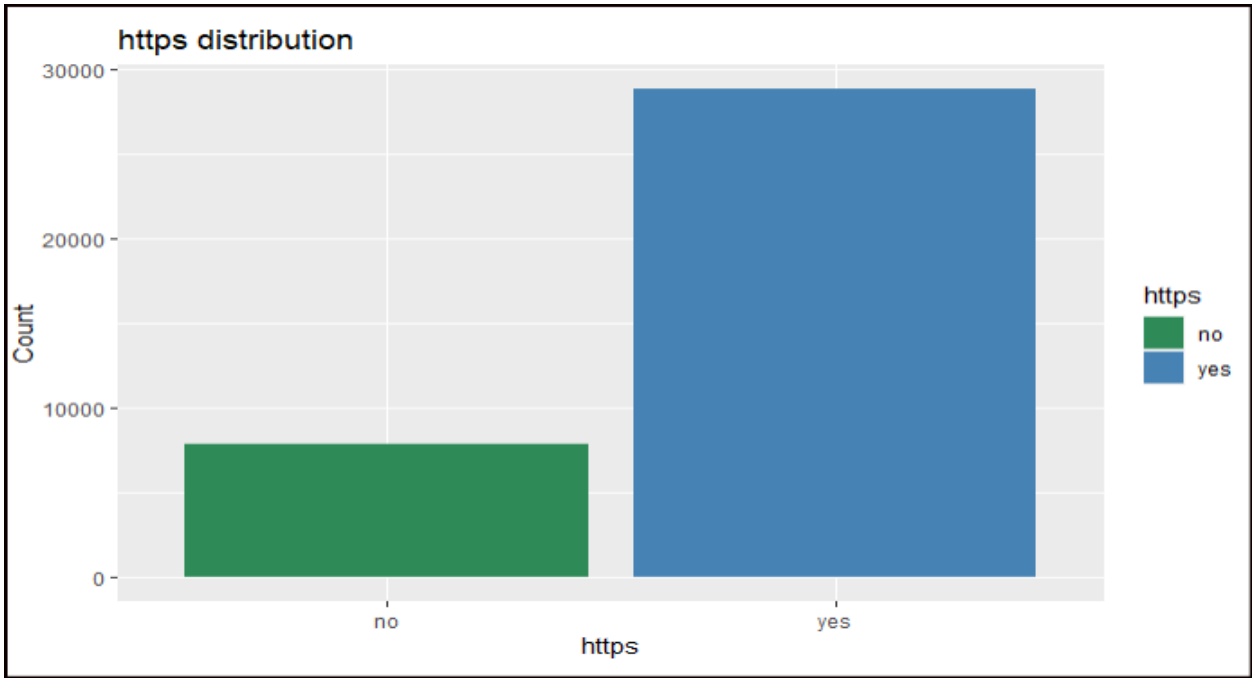
The objective of this case study is to create an algorithm that can efficiently and automatically determine the status of a website through which we can efficiently blacklist bad websites. This will help reduce the likelihood of data breaches occurring in our organization.

VISUALIZATION:

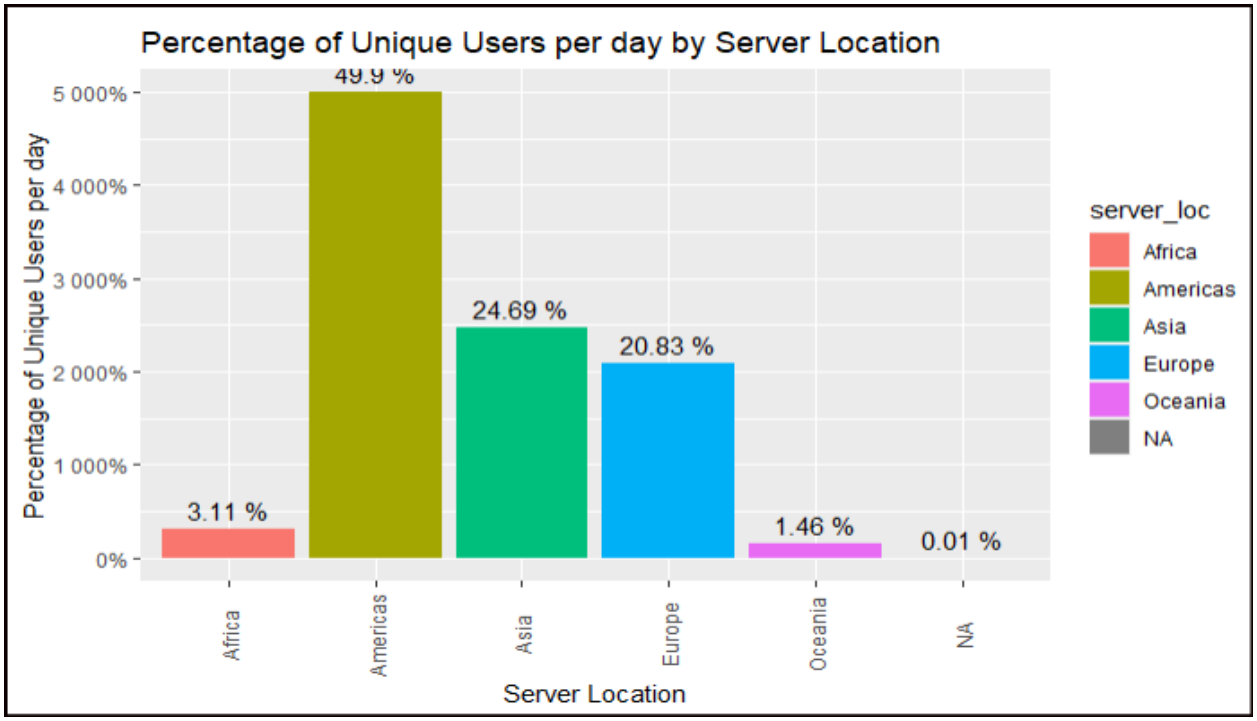
Plots drawn during the initial exploration of original labeled data are shown below:



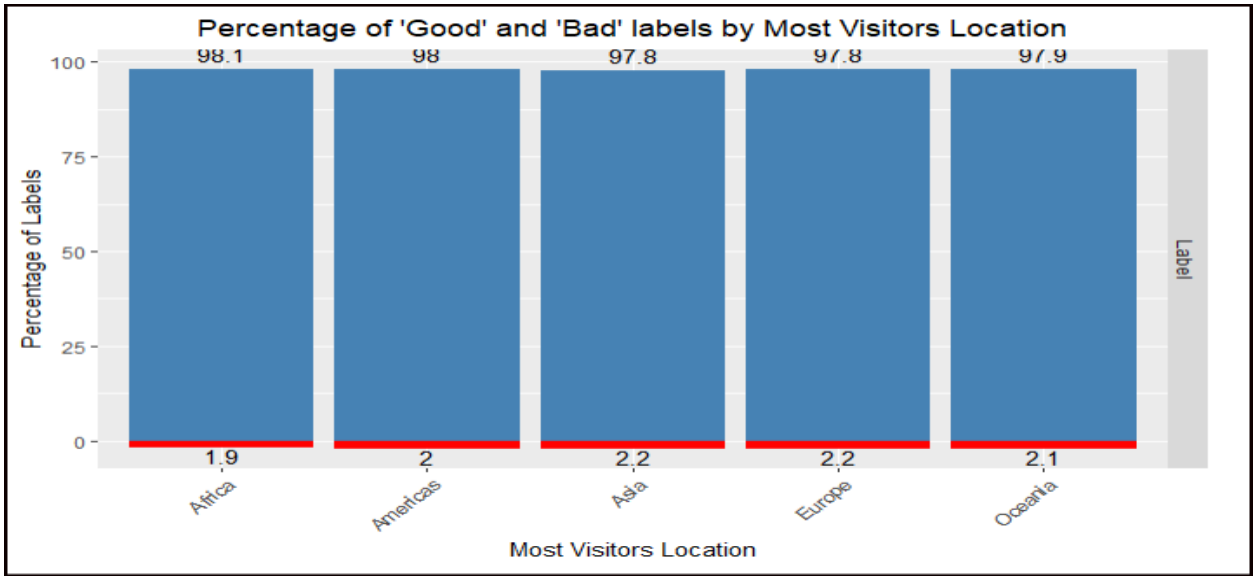
The number of registered domains are way higher than the number of unregistered domains in the original labeled dataset.



Websites using https in their URL are more compared to those which are not using it.



Percentage of unique users per day by Server Location are shown in the above plot. Here, we can observe that America's server location has almost half of the users count making it the highest of all.



Above plot shows the Percentage of 'Good' and 'Bad' labels by Most Visitors Location. All the locations have around 98% of good labels and only 2% of bad labels.

DATA PREPARATION:

The data preparation step involves loading the dataset, examining the structure and summary statistics of the data, handling missing values, and encoding categorical variables.

The first step was to load the original train dataset 'websites_labelled.df' and split it into a 75% training set and a 25% test set. A copy of the train and test sets was saved, and data preprocessing steps were performed on them. All the columns were checked for missing values, and the missing values in the Server_loc column were filled using the mode (most frequent) value. Categorical variables were converted into numerics using one-hot encoding and label encoding methods. The target variable's datatype was changed to factor.

MODEL TRAINING AND EVALUATION:

The model training and evaluation step involves selecting the features to be used, specifying hyperparameters, training and testing the model, and evaluating its performance using the confusion matrix.

Using the modified train and test sets, Naive Bayes, Random Forest, and Decision Tree algorithms were implemented. The three models were trained using the modified train data at first and then using feature importance technique followed by balanced train data obtained by applying SMOTE sampling. In addition, Hyperparameter tuning was performed to obtain the best-tuned model. Finally, a final model was built by selecting the best parameters from the above analysis. Evaluate the performance of the model using the confusion matrix.

After obtaining the accuracy for the three algorithms for each of the tasks mentioned above, the Naive Bayes model using the modified train data produced the best accuracy of 99.84%.

PSEUDO-LABELING:

The pseudo-labeling step involves using the trained model to predict labels for the unlabelled dataset, combining the labelled and unlabelled datasets, and retraining the model on the combined dataset.

The original test data provided 'websites_unlabelled.df' was loaded, and the same data preprocessing steps were performed on it. The best model obtained from the above analysis was applied to this original unlabelled data, and the labels were predicted.

CONCLUSION:

In summary, this case study demonstrates the development of an algorithm that can automatically determine the status of a website and blacklist bad websites, reducing the likelihood of data breaches. The Naive Bayes model run on modified train data produced the best accuracy, and this best model was applied to the original unlabelled data to predict the labels.