Marketing Campaign

Name:	**SHINILA M F**
Registration No./Roll No.:	21337
Institute/University Name:	IISER Bhopal
Program/Stream:	e.g., EECS
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

The objective of the project is to develop supervised machine learning frameworks to predict the success or failure of the campaigning based on customer’s response for a company. The dataset contains information about Marketing campaign of company data on customer profiles, product preferences, etc.The training dataset has 2016 rows and 25 columns whereas test dataset has 226 rows and 25 columns. Dataset have 2 classes; ’0’ for ’failure’ and ’1’ for ’success’ .1715 training instances belong to individual classes ‘0’ of dataset and 301 training instances belong to individual classes ‘1’ of dataset.**??**.

2 Methods

2.1 Read data

Python libraries are imported .Csv training and class label data set files are read and the dataframe are stored in df and df1 respectfully.

```
df = pd.read-csv(’marketing trn data.csv’).
```

2.2 Sanity check

Explore data set to gain insight of data set. No of rows and columns are known for both the dataset ”df.shape” and the training instances of individual classes ’1’ and ’0’ where’0’ for ’failure’ and ’1’ for ’success’ of marketing campaign.

```
df.info()
df.columns
df.dtypes
```

2.3 Reprocessing

Cleaning and processing of data is done to handle missing values,out liners etc. Feature engineering used to Identifying relevant features that contribute to campaign success. Dt Customer column is dropped .

```
df = df.drop(’Dt Customer’, axis=1)
```

On processing it is found that 3 columns are object type ’Education’, ’Marital Status’, ’ Income ’. First ’Income’ column is converted from string to integer by dropping and replacing the dollar and comma with blank space .

```
df[’ Income ’]=df[” Income ”].str.replace(’[$ ]’,””)
df[’ Income ’] = df[’ Income ’].astype(float)
df[’ Income ’] = pd.to numeric(df[’ Income ’], errors=’coerce’).astype(’Int64’)
```

Object is converted to float and then to integer .isnull() used to find the missing values of Income column.

for column in df.columns:

print(column.capitalize() + ' - Missing Values: ' + str(sum(df[column].isnull())))

Deep maps are used to visualise the same.

sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')

Mean of the income of different education '2n Cycle', 'Basic','Graduation','Master','PhD' are imputed in respective missing columns. One hot encoding used for the column 'Education', 'Marital Status' to convert categorical values hence features are increased along with columns .

df = pd.get dummies(df, columns = ['Education', 'Marital Status']) print(df)

Numerical variables are obtained hence discrete values are more than continuous variable .Hence it is a classification problem.

discrete feature=[feature for feature in numerical features if len(df[feature].unique())¡25]

print("Discrete Variables Count: ".format(len(discrete feature)))

Univariate distribution visualise the data which are normalized .

Correlation used to check the relation between the various features.Class label data is read .Data doesn't have a header.Header is created and a column is dropped and dataframe stored in labels.

df1.columns = ['Data points', 'Class lables']

df1.to csv('marketing trn class labels.csv', index=False)

labels = df1['Class lables']

print(df1)

2.4 Training and validification

Training and validification of data is done . Training and class labels are stored in x and y .They are split into X train, X test, y train, y test and Kflod used to split training set into 5 split.

skf = StratifiedKFold(shuffle=True, n splits=5, random state=123)

Splitting the dataset into training and validation sets to train the model and assess its performance on unseen data.Exploring various machine learning algorithms various model classifier are used to Evaluate models based on metrics such as accuracy, precision, recall, and F1 score.Fine-tuning hyperparameters to optimize the model's predictive capabilities.Classifers like desicion tree,logistic regression, KNN, Naivebayes, AdaBoost,SVM are used.

Decision tree with

param grid = 'criterion': ['gini', 'entropy'], 'max$_{d}epth' : [None, 10, 20, 30, 40, 50],' min_{s}amples_{s}plit' : [2, 5, 10],' min_{s}amples_{l}eaf' : [1, 2, 4]$

Random Forest Classifier

param grid rf = 'n$_{e}stimators' : [50, 100, 200],' max_{d}epth' : [None, 10, 20, 30],' min_{s}amples_{s}plit' : [2, 5, 10],' min_{s}amples_{l}eaf' : [1, 2, 4]$

K Neighbors Classifier

paramgrid knn = 'n neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'], 'p': [1, 2] 1 for Manhattan distance, 2 for Euclidean distance

Logistic Regression classifier param grid logreg = 'penalty': ['l1', 'l2'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000], 'solver': ['liblinear', 'saga']

Multinomial Naive Bayes classifer

param grid nb = 'alpha': [0.1, 0.5, 1.0, 2.0], 'fit$_{p}rior' : [True, False]$

Ada Boost Classifier

param grid adaboost = 'n$_{e}stimators' : [50, 100, 200],' learning_{r}ate' : [0.01, 0.1, 0.5, 1.0]$

Support Vector Machine classifier

param grid svm = 'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf', 'poly'], 'gamma': ['scale', 'auto']

github link

https://github.com/SHINILA/Project

3 Experimental Analysis

Random forest is the best classifier among all the model it shows the accuracy with 90 percentage and 345 correctly predicted as success ,8 belong to failure but predicted as success ,33 are correctly predicted as failure whereas 18 belong to success but are predicted as failure.Macro-averaged precision, recall, f-measure for a classification problem, or normalized mutual information, f-measure for a unsupervised problem are written in the following table.

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F-measure
Adaptive Boosting	0.74	0.74	0.74
Decision Tree	0.66	0.66	0.66
K-Nearest Neighbor	0.69	0.57	0.58
Logistic Regression	0.79	0.72	0.74
Random Forest	0.80	0.67	0.71
Naive baye	0.57	0.66	0.54
Support Vector Machine	0.68	0.68	0.68

Table 2: Confusion Matrices of Different Classifiers

Actual	Predicted Class	
Class	Success	Failure
Success	330	23
Failure	23	28

Adaptive Boosting

Actual	Predicted Class	
Class	Success	Failure
Success	324	29
Failure	31	20

Decision Tree

Actual	Predicted Class	
Class	Success	Failure
Success	345	8
Failure	43	8

K-Nearest Neighbor

Actual	Predicted Class	
Class	Success	Failure
Success	340	13
Failure	27	24

Logistic Regression

Actual	Predicted Class	
Class	Success	Failure
Success	345	8
Failure	33	18

Random Forest

Actual	Predicted Class	
Class	Success	Failure
Success	221	132
Failure	16	35

Multinomial Nive Bayes

Actual	Predicted Class	
Class	Success	Failure
Success	335	18
Failure	30	21

SVM

4 Discussion

This project successfully predicts the success and failure of the marketing campaign.This can be used to predict whether the new marketing strategy applied is successful or not and helps them to increase their profit and keep up with the trading in market .These has de merits as some failures are falsely predicted as success which lead to a loss of the trade in market.On comparing performances and checking for best threshold the best suited target variable pre- diction classifier is selected. Thus running the selected model, the framework predicts whether the marketing campaign is successful or failure.Model acn be extended as various algorithm can be used to run to enhance the acuracy and tunning the hyperparameter ,which will help the markets with their marketing campaign are really sucess in market or not

References

https://www.geeksforgeeks.org/
https://amplitude.com/blog/marketing-forecasting
https://www.javatpoint.com/machine-learning-prediction
https://www.linkedin.com/pulse/predicting-success-marketing-campaigns-using- machine-learning