**Introduction**

A magazine firm, like any other, is seeking to figure out who their regular subscribers are. The company's revenues have been declining, and we're attempting to figure out why. In addition, logistic regression and support vector machine methodologies are being used in this research. The dataset comprises 2240 rows and 29 columns. The dataset is dominated by two categorical variables: education and marital status. The rest of the information is either numerical or binary.

**Data cleaning**

The data has 2240 rows and 29 columns. In the data we don’t have many inconsistencies like values that are hindering the modeling process. We do have missing values.

Table

Description automatically generated

Figure1Missing Values

As we can see Income has 24 missing values that need to be dealt with. Here

we drop these missing values as they hold a small percentage of the entire data

sent.

**EDA**

In this section we talk about the exploratory data analysis that allows us to get a better understanding of the story the data is telling us and how different factors affect why people are not subscribing to magazines. To understand how the data is talking to us we evaluate the descriptive statistics of the data.

Graphical user interface, application

Description automatically generated

Figure 2 Descriptive Statistics

This allows us to understand how the data is trending and what are min max mean etc. of the data.

Chart, bar chart

Description automatically generated

Figure 3 Income v/s Education

He we have a Bar plot that takes about the number of subscribes based on the income and their education qualifications. This allows us to understand how these variables might affect the modeling and understanding of data later.

The creation of correlations plot allows us to see how different factors affect the and cause changes in target variable.

Graphical user interface, application, table, Excel

Description automatically generated

Figure 4 Correlation table

With the help of this correlation table, we can understand how strong or weak the relationship is between the dependent and the independent variables. The correlation graph makes it easier for us to understand the relation as with the help of the color code tend of the data is made easy.

Chart

Description automatically generated

Figure 5 Correlation plot

The correlation plot allows us to see how these variable trends are and what need to be considered for further analysis.

**Analysis**

In our analysis we are using VIF computational property that allows us to figure the variables that are influenced by multi-collinearity as multi-collinearity adversely affects the model and gives us incorrect outputs. We create dummies for education and marital status to give a better understanding of how these categories influence the model individually.

Text

Description automatically generated

Figure 6 VIF 1

With our initial VIF we see that there are few values that have a VIF of infinity which me they are perfectly collinear hence we drop 2 values from each encoded column. Here we drop ‘Education\_Basic’ and ‘Marital\_Status\_YOLO’. After this we get the following VIFs

Text

Description automatically generatedText, table

Description automatically generated

Figure 7 VIF 2

Here we see that there are multiple values that are higher than 100 so we drop those values. We drop the columns, ‘Marital\_Status\_Divorced’ , ‘Marital\_Status\_Married’, ‘Marital\_Status\_Single’, ‘Marital\_Status\_Together’, ‘Year\_Birth’. After this we compile an another VIF where we get the following:

Text

Description automatically generatedText, table

Description automatically generated

Figure 8 VIF 3

In this we still have high values that are above 10 so we drop those values that are above 10. The columns ‘Education\_Graduation’, ‘Income’ is dropped after which we compute VIF for the remaining values.

Text

Description automatically generated

Figure 9 VIF 4

Here, we drop all the columns that have a VIF of more than 5 that are columns ‘NumStorePurchases’, ‘NumWebVisitsMonth’, ‘NumWebPurchases’ with which we get the following final VIF.

Text

Description automatically generated

Figure 10 VIF 5

**Modeling:**

For our modeling, we are using Logistic Regression and Support Vector Machine(SVM). We set the train and test size to be split into 75:25.

**Logistic Regression:**

Logistic regression works on binary classification. We have our target variable to be AcceptedCmp1 which talks about the whether the customer accepted the campaign in the first campaign. With computation of the model, we get a score of 0.95307, a precision of 0.556 and a recall of 0.185. The precision tells us that 56.6 % of people will think whether to subscribe or not and 18.5 % will subscribe to the magazine. We compute a confusion matrix A screenshot of a computer

Description automatically generated with low confidence

Figure 11 Confusion Matrix Logistic

As per the confusion matrix, we see that 523 observations come in True Positive, and 4 observations come in False Positive. 22 observations are in False Negative and 5 are in True Negative.

**SVM :**

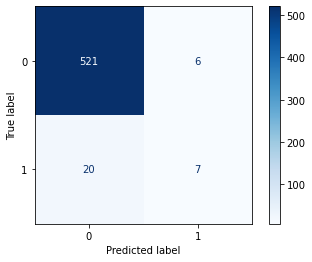
The linear Kernel contributes to the addition of extra dimensions for improved categorization of the Response variable. Interpreting the above-mentioned confusion matrix of an SVM model. The model's score is 0.9530. 

Figure 12 Confusion Matrix SVM

We discovered that 521 observations were True Positives 6 True Negatives. There were 7 false negatives and 20 True negatives among the findings. Here we have a precision of 0.5384 and a recall of 0.25925. The precision tells us that 53.84% of people will think, whether to subscribe or not and the recall tells us that 25.925 % will subscribe to the magazine.

**Conclusion**

* To conclude I would say that children being at home being at home which includes (kids and Teens) as people with children would tend to buy books that driven towards the education of those children.
* The people buying catalogs is another factor as people who buy a catalog already have a brief idea of what the magazine is about hence aren’t purchasing the magazine.
* Finally, the people who hold higher degrees like a master’s degrees would not be interested in things like fruits meats as the entire industry shifted to digital marketing when the pandemic hit.

Benchmarking Metrics:

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Score | Precision | Recall |
| Logistic Regression | 0.95307 | 0.5556 | 0.1851 |
| SVM | 0.95306 | 0.5384 | 0.2592 |

Here we see that logistic regression is better at classification in this case compared to SVM as it has a better score and Precision where SVM has a better recall.

**Reference:**

* Choosing colormaps in matplotlib#. Choosing Colormaps in Matplotlib - Matplotlib 3.6.2 documentation. (n.d.). Retrieved November 20, 2022, from https://matplotlib.org/stable/tutorials/colors/colormaps.html Zach. (2021, October 21).
* A guide to multicollinearity &amp; VIF in regression. Statology. Retrieved November 13, 2022, from <https://www.statology.org/multicollinearity-regression/>
* Sklearn.svm.SVC. scikit. (n.d.). Retrieved November 20, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
* Sklearn.linear\_model.logisticregression. scikit. (n.d.). Retrieved November 20, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

**Appendix 1**

**Code:**

# In[2]:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix,plot\_confusion\_matrix

from sklearn import metrics

from sklearn import svm

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from sklearn.model\_selection import train\_test\_split

import warnings

import statsmodels.api as sm

warnings.filterwarnings('ignore')

# In[3]:

data = pd.read\_csv("C:/Users/kgrat/OneDrive/Documents/ALY 6020/marketing\_campaign.csv")

# In[4]:

data.info()

# In[5]:

data.shape

# In[6]:

data['AcceptedCmp1'].value\_counts()

# In[7]:

data.isna().count()

# In[8]:

print('sum of null values: {}'.format(data.isnull().sum()))

# In[9]:

data = data.dropna()

# In[10]:

data.describe()

# In[11]:

sns.barplot(x='Education',y='Income', data=data,

hue='Response')

# In[12]:

data.head(5)

# In[13]:

data.columns

# In[14]:

data = data.drop(['AcceptedCmp3','AcceptedCmp4','AcceptedCmp5','AcceptedCmp2','Z\_CostContact','Z\_Revenue','Response','Dt\_Customer'],axis=1)

# In[15]:

data.info()

# In[16]:

y = data.pop('AcceptedCmp1')

data.insert(0,'AcceptedCmp1',y)

#In[17]:

c= data.corr()

c

# In[18]:

mask = np.triu(np.ones\_like(data.corr(), dtype=bool))

# In[19]:

plt.figure(figsize=((14,14)))

sns.heatmap(c,annot=True, mask=mask, vmin=-1, vmax=1)

# In[20]:

data

# In[21]:

data =pd.get\_dummies(data,columns=['Education','Marital\_Status'])

y = data['AcceptedCmp1']

data = data.drop('AcceptedCmp1', axis= 1)

# In[22]:

temp = pd.Series([variance\_inflation\_factor(data.values, i) for i in range (data.shape[1])], index = data.columns)

temp

# In[23]:

data = data.drop(['Education\_Basic','Marital\_Status\_YOLO'], axis= 1)

# In[24]:

temp = pd.Series([variance\_inflation\_factor(data.values, i) for i in range (data.shape[1])], index = data.columns)

temp

# In[25]:

data = data.drop(['Marital\_Status\_Divorced','Marital\_Status\_Married','Marital\_Status\_Single','Marital\_Status\_Together','Year\_Birth'],axis= 1)

# In[26]:

temp = pd.Series([variance\_inflation\_factor(data.values, i) for i in range (data.shape[1])], index = data.columns)

temp

# In[27]:

data = data.drop(['Education\_Graduation','Income'],axis= 1)

# In[28]:

temp = pd.Series([variance\_inflation\_factor(data.values, i) for i in range (data.shape[1])], index = data.columns)

temp

# In[29]:

data = data.drop (['NumStorePurchases','NumWebVisitsMonth','NumWebPurchases'],axis=1)

# In[30]:

temp = pd.Series([variance\_inflation\_factor(data.values, i) for i in range (data.shape[1])], index = data.columns)

temp

# In[31]:

x = data

# In[32]:

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=.25, random\_state=123)

# # Logistic Regresion

# In[33]:

logistic\_regression= LogisticRegression(solver= 'lbfgs',max\_iter=2000)

model=logistic\_regression.fit(x\_train,y\_train)

# In[34]:

pred= model.predict(x\_test)

print('Accuracy of the model is:{:.5f}'.format(model.score(x\_test, y\_test)))

# In[35]:

print("Precision:",metrics.precision\_score(y\_test,pred))

print("Recall:",metrics.recall\_score(y\_test,pred))

# In[36]:

plot\_confusion\_matrix(logistic\_regression,x\_test, y\_test,cmap='Greens')

plt.show()

# # Support Vector Machine

# In[37]:

SVM = svm.SVC(kernel='linear')

# In[38]:

SVM.fit(x\_train,y\_train)

# In[39]:

svm\_pred = SVM.predict(x\_test)

# In[40]:

plot\_confusion\_matrix(SVM,x\_test,y\_test,cmap='Blues')

# In[41]:

SVM.score(x\_test, y\_test)

# In[42]:

print("Precision:",metrics.precision\_score(y\_test,svm\_pred))

print("Recall:",metrics.recall\_score(y\_test,svm\_pred))