Supervised Learning Classification Project

Submission by: Koto Andrew Omiloli – 01/11/2024
Please see notebook code and for project output results

1. The main objective of this analysis is to generate outcomes of customers in Walmart Super Store, that are likely to churn or not given the following features `accountlength', 'internationalplan', 'voicemailplan', 'numbervmailmessages', 'totaldayminutes', 'totaldaycalls', 'totaldaycharge', 'totaleveminutes', 'totalevecalls', 'totalevecharge', 'totalnightminutes', 'totalnightcalls', 'totalnightcharge', 'totalintlminutes', 'totalintlcalls', 'totalintlcharge' and 'numbercustomerservicecalls'.`

With this analysis the Company will be able to know beforehand the likelihood a particular customer will churn or not so as to predict their total customers, profits and know which service to improve in order to retain more customers.

2. This data set gives the details of customers that churn or did not in Walmart Super Store. It contains 5000 instances, and 7 features. The target variable is 'churn' of type object. The objective is to explore the data by checking if it has a balanced class – if it is unbalanced, stratified split is used to generate equal proportionality with our train and test set (see notebook attached), label encoding the outcome variable, carry out a train and test split, and using a linear classifier, SVM and … to predict the outcome across different hypermeter to get the right balance between bias and variance.

3. Step 1: The data was inspected and what was discovered was two features where objects i.e 'internationalplan' and 'voicemail' because the features where required to be float types to perform logisticregression, I took the decision to drop those features out by using the command

```
feature_cols=[x for x in data.columns if x not in 'churn &
internationalplan and voicemailplan']
```

4. Step 2: Checking for skewed features inorder to apply log transformation. The output from the code below shows 6 features required log transformation, they were as follows

Features	Skew values
totalintlcalls	1.360692
numbervmailmessages	1.350493
numbercustomerservicecalls	1.042462
totalintlminutes	-0.209966
totalintlcharge	-0.210286

dtype: float64

#Log transforming skew variables - feature engineering

```
#Creating a list of float columsn to check for skewing
num_cols = df.select_dtypes('number').columns
```

```
#This is the max limit above which log tranform will be carried out
skew_limit = 0.2
skew_vals = df.skew()

#To show the skewed columns
skew_cols =
skew_vals[abs(skew_vals)>skew_limit].sort_values(ascending=False)
skew_cols

fig, ax1 = plt.subplots(1,2)
field1 ="totalnightminutes"
df["totalnightminutes"].apply(np.log1p).hist(ax=ax1[0])
```

Secondly, I used stratified train and test split in order to create a balance between the (x_train,y_train and x_test,y_test).

5. The prediction was carried out across two base line models i.e Logisticregression and GradientBoosting Classifier and then a VotingClassifier model using the 'soft' approach was used in averaging the probabilities of the base line models mentioned as shown in the Jupiter notebook. A sample of code is shown here as well

```
#Using a stackmodel - logisticregression and GradienBoostClassifier
from sklearn.ensemble import VotingClassifier
estimators =[('LR', lr), ('LR_L2', lr_l1), ('GBC', GBC)]
VC = VotingClassifier(estimators,  voting='soft')
VC =VC.fit(x_train,y_train)

#predicting using VC
y_pred =VC.predict(x_test)
print(classification_report(y_test,y_pred))

#confusion matrix visualization
cm = confusion_matrix(y_test,y_pred)
sns.heatmap(cm, annot=True )
```

6. Based on the precision, recall, f1-score and support scores of the different models deployed, the stacked model had the most impressive performance results. it had a recall value of 1.00 meaning it had high correct minority classification. The stacked model had equal precision and recall value of 0.87 similar to the individual models developed. Hence, it is my recommendation that the stacked model be deployed in predicting the number of customers that will churn or not in Walmart Super Store

Summary of key results of the models used in this project

Logistic Regression lr model

	precision	recall	f1-score	support
0	0.87	0.99	0.93	1717
1	0.68	0.11	0.19	283
accuracy			0.87	2000
macro avg	0.78	0.55	0.56	2000
weighted avg	0.84	0.87	0.82	2000

LogisticRegression lr l1 Model

	precision	recall	f1-score	support

```
              precision    recall  f1-score   support

           0       0.88      0.99      0.93      1717
           1       0.68      0.14      0.24       283

    accuracy                           0.87      2000
   macro avg       0.78      0.57      0.58      2000
weighted avg       0.85      0.87      0.83      2000
```

GradientBoosting Classifier

```
Fitting model with 15 trees
              precision    recall  f1-score   support

           0       0.89      0.99      0.94      1717
           1       0.88      0.25      0.39       283

    accuracy                           0.89      2000
   macro avg       0.88      0.62      0.67      2000
weighted avg       0.89      0.89      0.86      2000
```

VotingClassifier Model

```
precision    recall  f1-score   support

           0       0.87      1.00      0.93      1717
           1       0.91      0.11      0.20       283

    accuracy                           0.87      2000
   macro avg       0.89      0.55      0.56      2000
weighted avg       0.88      0.87      0.83      2000
```

7. In the future, I would recommend further addition of other models such as support vectors machine (SVM) and AdaBoost Classifier and use GridsearchCV to run a cross validation and then use that best performing output for each classifier in a stack model so that a voting classifier can be used to average the probabilities to get an overall best model to used for this project prediction in Walmart Super Store.