Project :

Advance Forecasting Model to predict Subscription Cancelation

Tarun Sharma

Machine Learning Engineer

To produce the best hyperparameters and ensemble models for our optimized Logistic Regression model and Naïve Bayes models

Tools Used:
Jupyter Notebook (Python with Pandas, NumPy, Matplotlib)

Advance modeling

Using Isolation Forest to remove outliers

Using SMOTE for Class imbalance correction

Learning Curve analysis to find best model out of two

Using K-folds method to find best hyperparameters for optimized model.

Ensemble Models to use Voting, Stacking & Bagging algorithms to find combination of LR and NB models to give us the best combined model for predictions

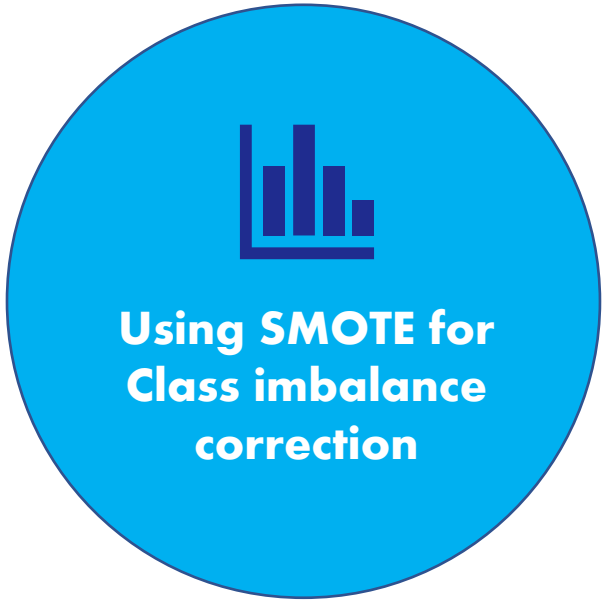
Advance modeling


```
# Use built-in isolation forest
from sklearn.ensemble import IsolationForest

# The prediction returns 1 if sample point is inlier. If outlier prediction returns -1
clf_all_features = IsolationForest(random_state=100)
clf_all_features.fit(x_train)

#Predict if a particular sample is an outlier using all features for higher dimensional data set.
y_pred_train = clf_all_features.predict(x_train)
y_pred_train2 =np.array(list(map(lambda x: x == 1, y_pred_train)))

# Exclude suggested outlier samples for improvement of prediction power/score
x_train_mod = x_train[y_pred_train2, ]
y_train_mod = y_train[y_pred_train2, ]

#Size of Datasets
print('Original Train Dataset Size : {}'.format(len(x_train)))
print('New Train Dataset Size      : {}'.format(len(x_train_mod)))
```

```
Original Train Dataset Size : 2666
New Train Dataset Size      : 2124
```

552 outliers have been removed

Advance modeling

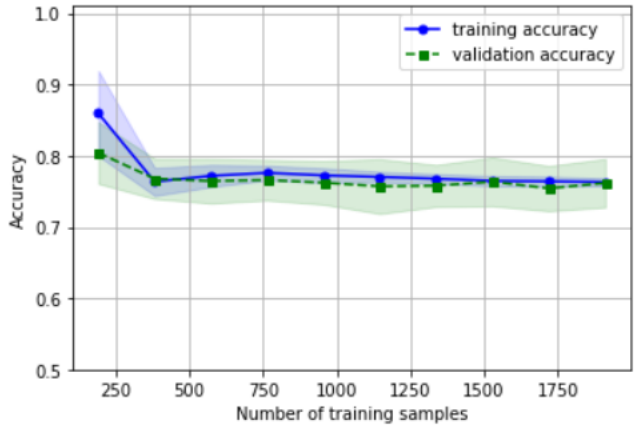



```
#Fix the imbalanced Classes
from imblearn.over_sampling import SMOTE
smt=SMOTE(random_state=100)
x_train_smt,y_train_smt = smt.fit_resample(x_train_mod,y_train_mod)

#Scale the Data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train2 = sc.fit_transform(x_train_smt)
x_test2 = sc.fit_transform(x_test)

x_2 = sc.fit_transform(x)

#Models
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
```

Learning Curve Analysis

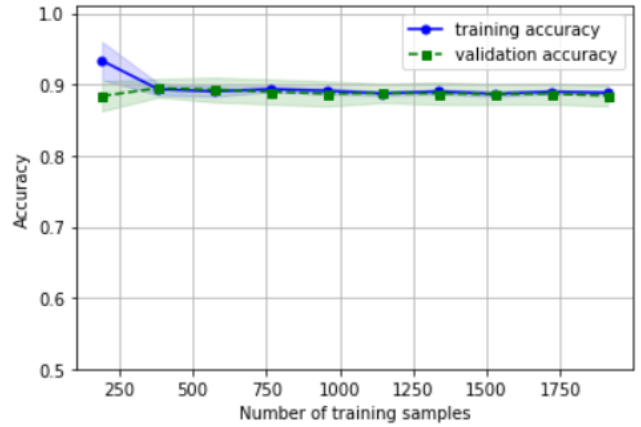
Learning Curve analysis to find best model out of two

Logistic Regression - Learning Curve



- The validation accuracy is around 0.76
- **Model is good at learning**, and its predictions are giving best accuracy of 0.78.
- Very Low Variance

Naive Bayes - Learning Curve



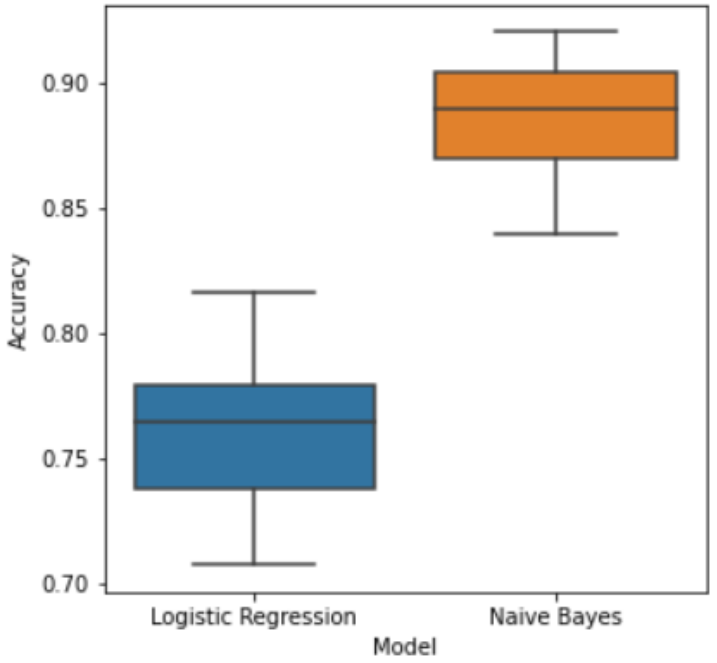
- The validation accuracy is around 0.88
- **Model is good at learning**, and its predictions are giving best accuracy of 0.88 which is better than LR
- Very Low Variance, High Bias and looks like over-fitting model

Naïve Bayes performed better than Logistic Regression

Advance modeling

Using K-folds method to find best Hyperparameters for optimized model.

```
Model Evaluation - Accuracy Score
Logistic Regression 0.76 +/- 0.03
Naive Bayes 0.89 +/- 0.02
```



- Average accuracy of LR model is 0.76 with an error of +/- 0.03
- Average accuracy of NB model is 0.89 with an error of +/- 0.02
- Accuracy for NB is going as high as 0.92 for one of the combination and going as low as 0.83 for some other.
- LR seems to do a worse job than NB at predicting.

Naïve Bayes performed better than Logistic Regression

Optimized Model Analysis

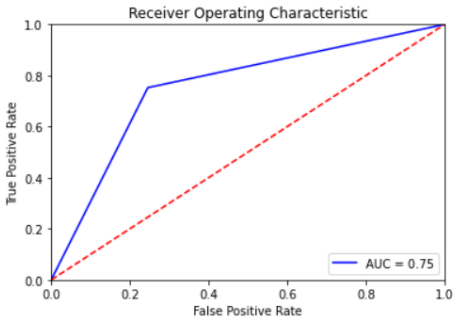
Model Name: LogisticRegression(class_weight='balanced', random_state=100)

Best Parameters: {'clf__C': 0.01, 'clf__penalty': 'l2'}

```
 [[430 140]
 [ 24  73]]
```

	precision	recall	f1-score	support
Outcome 0	0.95	0.75	0.84	570
Outcome 1	0.34	0.75	0.47	97
accuracy			0.75	667
macro avg	0.64	0.75	0.66	667
weighted avg	0.86	0.75	0.79	667

ROC Curve



- Model is very bad at predicting 'Subscription canceled'
- Model is comparatively better at predicting 'Subscription not canceled'
- Accuracy of 0.75 is quite average
- AUC = 0.75 is a fair model (not good not bad)

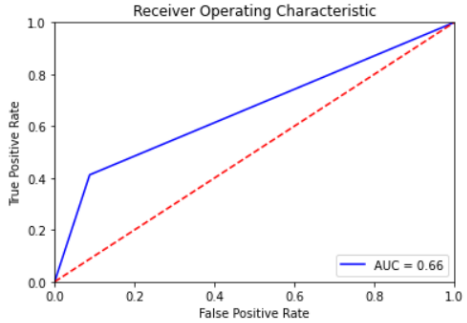
Model Name: GaussianNB()

Best Parameters: {}

```
 [[520  50]
 [ 57  40]]
```

	precision	recall	f1-score	support
Outcome 0	0.90	0.91	0.91	570
Outcome 1	0.44	0.41	0.43	97
accuracy			0.84	667
macro avg	0.67	0.66	0.67	667
weighted avg	0.83	0.84	0.84	667

ROC Curve



- Model is very bad at predicting 'Subscription canceled'
- False Positives is low i.e 50 when compared to LR
- False Negative is very high i.e 57
- Model is comparatively better at predicting 'Subscription not canceled'
- Accuracy of 0.84 is good
- AUC = 0.66 is a poor model (not good not bad)

LR has better AUC than NB, Both the models have under-performed

Advance modeling – Voting Model

Ensemble Models to use Voting, Stacking & Bagging algorithms to find combination of LR and NB models to give us the best combined model for predictions

```
#Create Voting Model - Sklearn
from sklearn.ensemble import VotingClassifier
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import cross_validate
from sklearn.ensemble import BaggingClassifier

estimators = []

model1 = LogisticRegression(solver='lbfgs',class_weight='balanced',
                            random_state=100)
estimators.append(('Logistic', model1))

model2 = GaussianNB()
estimators.append(('Naive Bayes', model2))

voting_clf=VotingClassifier(estimators,voting='soft')

scoring = {'acc': 'accuracy',
           'prec_macro': 'precision_macro',
           'rec_macro': 'recall_macro'}
print('\nVoting Model')
for clf in (model1,model2,voting_clf):
    rkfcv= clf.fit(x_train2,y_train_smt)
    ens_rkf1 = RepeatedKFold(n_splits=10, n_repeats=5, random_state=100)
    rKFcv = cross_validate(rkfcv, x_2, y, scoring=scoring, cv=ens_rkf1)
    print(clf.__class__.__name__,round(rKFcv['test_rec_macro'].mean(),2))
```

```
Voting Model
LogisticRegression 0.76
GaussianNB 0.67
VotingClassifier 0.72
```

Voting model has chosen Logistic Regression = 0.76
A combination of both algorithm (Average) that is NB and LR has a score of 0.72

Stacking and Bagging Classifier

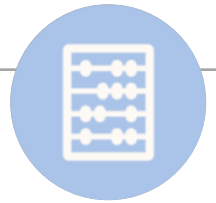
Ensemble Models to use Voting, Stacking & Bagging algorithms to find combination of LR and NB models to give us the best combined model for predictions

```
#Create Stacking Model-Sklearn
from sklearn.ensemble import StackingClassifier

#Identify Models
lr = LogisticRegression(solver='lbfgs',class_weight='balanced',
                        random_state=100)

estimators2 = []

mod1 = GaussianNB()
estimators2.append(('Naive Bayes', mod1))

mod2 = BaggingClassifier(random_state=100)
estimators2.append(('Bagging', mod2))

#Create Stacking Classifier
stackmod=StackingClassifier(estimators=estimators2,
                            final_estimator=lr)

scoring2 = {'acc': 'accuracy',
           'prec_macro': 'precision_macro',
           'rec_macro': 'recall_macro'}

print('\nStacking Model')
for clf in (mod1,mod2,stackmod):
    rkfcv2= clf.fit(x_train2,y_train_smt)
    ens_rkf2 = RepeatedKFold(n_splits=10, n_repeats=5, random_state=100)
    rKFcv2 = cross_validate(rkfcv2, x_2, y, scoring=scoring2, cv=ens_rkf2)
    print(clf.__class__.__name__,round(rKFcv2['test_rec_macro'].mean(),2))
```

```
Stacking Model
GaussianNB 0.67
BaggingClassifier 0.8
StackingClassifier 0.85
```

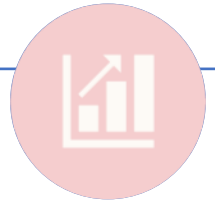
Bagging model has given good score = 0.8
Stacking model has given good score = 0.85
Stacking Classifier has performed better than Bagging and standalone Logistical Regression

Model Recommendation



USING LOGISTIC REGRESSION

Voting model gave the best score to Logistical Regression Model because the average of both model i.e 0.72 is less than individual score of LR i.e 0.76



USING STACKING MODEL

Stacking model has far better score i.e 0.85 than LR or Bagging Model

LABEL SET IMBALANCE

Label set for training the models is highly imbalanced i.e

Subscription Canceled - 483

Subscription not Canceled - 2850

.