Self Case Study - 1

Customer Relationship Prediction

In [1]:

```
!pip install -U scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (0.22.2.post1)
Collecting scikit-learn
  Downloading scikit_learn-0.24.2-cp37-cp37m-manylinux2010_x86_64.whl (22.3 MB)
     |████████████████████████████████| 22.3 MB 5.7 MB/s
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-lear
n) (1.0.1)
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.7/dist-packages (from scikit-lea
rn) (1.4.1)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.7/dist-packages (from scikit-lea
rn) (1.19.5)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-2.2.0-py3-none-any.whl (12 kB)
Installing collected packages: threadpoolctl, scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 0.22.2.post1
    Uninstalling scikit-learn-0.22.2.post1:
      Successfully uninstalled scikit-learn-0.22.2.post1
Successfully installed scikit-learn-0.24.2 threadpoolctl-2.2.0
```

In [91]:

```
import pickle
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt



from sklearn.metrics import roc_auc_score,roc_curve
from prettytable import PrettyTable
import time
```

In [3]:

```
import sys
import six
sys.modules['sklearn.externals.six'] = six
```

Loading Data

In [4]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/X_test.pickle', 'rb') as handle:
    X_test = pickle.load(handle)
columns = X_test.columns
```

In [5]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/y_test_appetency.pickle', 'rb') as handle:
    y_test_appetency = pickle.load(handle)

with open('/content/drive/MyDrive/Case Study 1/Data/final/y_test_churn.pickle', 'rb') as handle:
    y_test_churn = pickle.load(handle)

with open('/content/drive/MyDrive/Case Study 1/Data/final/y_test_upselling.pickle', 'rb') as handle:
```

```
    y_test_upselling = pickle.load(handle)
```

In [6]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/not_nan_features.pickle', 'rb') as handle:
    features = pickle.load(handle)
```

In [7]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/data_mean.pickle', 'rb') as handle:
    data_mean = pickle.load(handle)
```

In [8]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/ordinal_encoder_clustering.pickle', 'rb') as
handle:
    ordinal_enocoder_imputation_cluster = pickle.load(handle)
```

In [9]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/clusters.pickle', 'rb') as handle:
    kmeans_clusters = pickle.load(handle)
```

In [10]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/polynomail_featurizer.pickle', 'rb') as handl
e:
    polynomail_featurizer = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator P
olynomialFeatures from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code
or invalid results. Use at your own risk.
  UserWarning)
```

In [11]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/Feature engg/Appetency/feature_group_200.pickle', '
rb') as handle:
    feature_group_200 = pickle.load(handle)
```

In [12]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/Feature engg/Appetency/feature_group_50.pickle', 'r
b') as handle:
    feature_group_50 = pickle.load(handle)
```

In [13]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/average_per_category.pickle', 'rb') as handle
:
    average_per_category = pickle.load(handle)
```

In [14]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/cat_encodings.pickle', 'rb') as handle:
    cat_encodings = pickle.load(handle)
```

In [15]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/vanilla_standard_scaler.pickle', 'rb') as han
dle:
    vanilla_standard_scaler = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator S
tandardScaler from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or
invalid results. Use at your own risk.
  UserWarning)
```

In [16]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/poly_standard_scaler.pickle', 'rb') as handle
:
    poly_standard_scaler = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator S
tandardScaler from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or
invalid results. Use at your own risk.
  UserWarning)
```

In [17]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/appetency_model_final.pickle', 'rb') as handl
e:
    appetency_model_final = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator S
GDClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or i
nvalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator D
ecisionTreeClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator R
andomForestClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator L
abelEncoder from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or in
valid results. Use at your own risk.
  UserWarning)
```

In [18]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/churn_model_final.pickle', 'rb') as handle:
    churn_model_final = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator S
GDClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or i
nvalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator D
ecisionTreeClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator R
andomForestClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator L
abelEncoder from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or in
valid results. Use at your own risk.
  UserWarning)
```

In [19]:

```
with open('/content/drive/MyDrive/Case Study 1/Data/final/upselling_model_final.pickle', 'rb') as handl
e:
    upselling_model_final = pickle.load(handle)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator S
GDClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or i
nvalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator D
ecisionTreeClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator R
andomForestClassifier from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:315: UserWarning: Trying to unpickle estimator L
abelEncoder from version 0.22.2.post1 when using version 0.24.2. This might lead to breaking code or in
valid results. Use at your own risk.
  UserWarning)
```

Utility functions

In [20]:

```
def impute_data(X):
    '''This method takes raw data(in dataframe format) as input and tuple of imputed data. The imputed
data is returned as tuple of (numerical, categorical data)'''
    numerical_data = X.iloc[:,:42].fillna(data_mean)
    categorical_data = X.iloc[:,42:].fillna('Others')

    return (numerical_data, categorical_data)
```

In [21]:

```
def defining_clusters(X):
    '''This method takes raw data(in dataframe format) as input and returns clustering labels of the in
put'''
    clustering_labels = []
    # before imputing data, we need to take care of NaNs
    numerical_features, categorical_data = impute_data(X)

    # encoding categorical features
    ordinal_features = ordinal_enocoder_imputation_cluster.transform(categorical_data)
    final_features = np.hstack((numerical_features, ordinal_features))

    # applying clustering
    for cluster in kmeans_clusters:
        clustering_labels.append(cluster.predict(final_features))
    return clustering_labels
```

In [22]:

```
def featurize(X, clustering_labels):
    '''This feature take raw data and clustering_labels as input and returns new featurized data.'''

    # 1.Count of NaN in datapoint
    X['NaN_Count'] = X.isna().sum(axis = 1)

    # 2.Binary indicator for NaN in a feature
    for column in X.columns[0:-1]:
        X[column+'_NaN'] = np.where(X[column].isna(),1,0)

    # 3. Cluster label
    for i in range(len(clustering_labels)):
        X['Cluster_'+str(i)] = clustering_labels[i]

    # 4. Polynomial Features

    ## Before generating polynomial features, we need to impute data.
    numerical_data, categorical_data =  impute_data(X)
    categorical_data = categorical_data.iloc[:,:30]

    X.iloc[:,42:72] = categorical_data
```

```
    X_nan_count = X['NaN_Count'].values.reshape(-1,1)

    X_clusters = X.iloc[:,-5:].values

    X_binary_ind = X.iloc[:,73:145].values

    X_polynomail_features = polynomail_featurizer.transform(numerical_data)


    # 5. Average value for feature group
    ## Feature group with mean under or close to 50
    X['feature_group_50_mean'] = X[feature_group_50].mean(axis =1)

    ## Feature group with mean under or close to 200
    X['feature_group_200_mean'] = X[feature_group_200].mean(axis =1)

    X_feature_means = X.iloc[:,-2:].values


    # 6. Average value per category
    categorical_cols = X.iloc[:,42:72].columns

    for col in range(len(categorical_cols)):
        X = pd.merge(X,average_per_category[col].iloc[:,:42],how = 'left', on = categorical_cols[col],
suffixes = (None, '_'+categorical_cols[col]+'_avg'))

    ## imputing the NaNs generated during last featurization
    for col in X.iloc[:,152:].columns:
        mean_col = col.split('_')[0]
        X[col] = np.where(X[col].isna(), data_mean[mean_col], X[col])

    X_averages =  X.iloc[:,152:].values


    # Categorical encoding - freq encoding
    for col in range(len(categorical_cols)):
        X.loc[:,categorical_cols[col]+'_encoding'] = X[categorical_cols[col]].map(cat_encodings[col])

    X.fillna(0,inplace = True)

    X_cat_encoding = X.iloc[:,-30:].values

    X_vanilla = np.hstack([numerical_data,X_cat_encoding])

    X_poly = np.hstack((X_polynomail_features, X_nan_count, X_clusters, X_feature_means, X_averages, X_
cat_encoding))

    X_vanilla = vanilla_standard_scaler.transform(X_vanilla)

    X_poly = poly_standard_scaler.transform(X_poly)

    X_poly = np.hstack((X_poly, X_binary_ind))

    return X_vanilla,X_poly
```

In [23]:

```
def preprocess(X):
    '''This function takes raw data as input and returns preprocessed data for modeling.'''
    # coverting to dataframe for ease of operations
    X =pd.DataFrame(data = X, columns= columns)

    # Removing features having NaN count more than 60% of total data.
    X =  X.iloc[:,features]

    # generating clustering labels
    clustering_labels = defining_clusters(X)

    # dropping duplicate columns
    X =X.drop(['Var220','Var222'], axis = 1)

    # featurizing dataset
    X_vanilla, X_poly = featurize(X,clustering_labels)

    return X_vanilla, X_poly
```

In [97]:

```
X = X_test.to_numpy()
```

In [98]:

```
X.shape
```

Out[98]:

```
(10000, 230)
```

Final Function 1

In [99]:

```
def predict(X_vanilla, X_poly):
    '''This function takes in 2 dataset input: vanilla and poly and predicts appetency, churn and upsel
ling '''
    y_appetency_pred = appetency_model_final.predict(X_poly)
    y_churn_pred =  churn_model_final.predict(X_poly)
    y_upselling_pred = upselling_model_final.predict(X_vanilla)

    return y_appetency_pred, y_churn_pred, y_upselling_pred
```

In [100]:

```
def final(X):
    '''This method takes raw data X as input as numpy array and returns prediction as output'''
    X_vanilla, X_poly = preprocess(X)

    appetency_pred, churn_pred, upselling_pred = predict(X_vanilla, X_poly)
    return appetency_pred, churn_pred, upselling_pred
```

In [101]:

```
%%time
y_appetency_pred, y_churn_pred, y_upselling_pred =  final(X)
```

```
CPU times: user 8.73 s, sys: 538 ms, total: 9.27 s
Wall time: 7.75 s
```

Final Function 2

In [102]:

```
def predict(X_vanilla, X_poly, y_test_appetency, y_test_churn, y_test_upselling):
    '''This function takes in 2 dataset and class labels and calulates and retuns mean and individual A
UC metric'''
    y_appetency_pred = appetency_model_final.predict_proba(X_poly)[:,1]
    y_churn_pred =  churn_model_final.predict_proba(X_poly)[:,1]
    y_upselling_pred = upselling_model_final.predict_proba(X_vanilla)[:,1]

    auc_score_appetency = roc_auc_score(y_test_appetency, y_appetency_pred)
    auc_score_churn = roc_auc_score(y_test_churn, y_churn_pred)
    auc_score_upselling = roc_auc_score(y_test_upselling, y_upselling_pred)

    mean_auc = (auc_score_appetency + auc_score_churn + auc_score_upselling) / 3

    return auc_score_appetency, auc_score_churn, auc_score_upselling, mean_auc
```

In [103]:

```
def final(X,y_appetency,y_churn,y_upselling):
    '''This function takes data in input X as numpy array and returns auc score for each label and mean auc score as output'''
    X_vanilla, X_poly = preprocess(X)

    auc_score_appetency, auc_score_churn, auc_score_upselling, mean_auc = predict(X_vanilla, X_poly, y_appetency, y_churn, y_upselling)
    return auc_score_appetency, auc_score_churn, auc_score_upselling, mean_auc
```

In [104]:

```
%%time
auc_score_appetency,auc_score_churn,auc_score_upselling,mean_auc =  final(X_test, y_test_appetency, y_test_churn, y_test_upselling)
```

```
CPU times: user 8.41 s, sys: 591 ms, total: 9.01 s
Wall time: 7.44 s
```

In [105]:

```
x = PrettyTable()
x.field_names = ["Label", "AUC Score",]
x.add_row(["Appetency", auc_score_appetency])
x.add_row(["Churn", auc_score_churn])
x.add_row(["Upselling", auc_score_upselling])
x.add_row(["Mean Score ", mean_auc])
```

In [106]:

```
print(x)
```

```
+-------------+--------------------+
|    Label    |     AUC Score      |
+-------------+--------------------+
|  Appetency  | 0.8700550535575369 |
|    Churn    | 0.7569458185363356 |
|  Upselling  | 0.8857708018142303 |
| Mean Score  | 0.8375905579693675 |
+-------------+--------------------+
```