Importing libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from imblearn.under_sampling import NearMiss
from imblearn.over_sampling import SMOTE

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report,ConfusionMatrixDisplay
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC
import xgboost as xgb
```

importing data

```
data = pd.read_csv('/content/Cust_Churn.csv')
```

```
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mult
0	7590-VHVEG	Female	0	Yes	No	1	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	
3	7795-CFOCW	Male	0	No	No	45	No	
4	9237-HQITU	Female	0	No	No	2	Yes	

5 rows × 21 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
```

```
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
data = data.replace(' ', value=0)
data.TotalCharges = pd.to_numeric(data.TotalCharges)
data.TotalCharges.dtype
```

```
dtype('float64')
```

```
g = sns.countplot(x="Churn",data=data, palette="muted")
g.set_ylabel("Customers", fontsize=14)
g.set_xlabel("Churn", fontsize=14)

data.Churn.value_counts(normalize=True)
```

```
No     0.73463
Yes    0.26537
Name: Churn, dtype: float64
```

Preprocessing

```
data = data.drop('customerID', axis = 1)
```

```
data.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	I
0	Female	0	Yes	No	1	No	No phone service	
1	Male	0	No	No	34	Yes	No	
2	Male	0	No	No	2	Yes	No	
3	Male	0	No	No	45	No	No phone service	
4	Female	0	No	No	2	Yes	No	

```
catagorical = [i for i in data.columns if data[i].dtypes == 'object']

for i in catagorical:
    print(i, ':', data[i].unique())
```

```
gender : ['Female' 'Male']
Partner : ['Yes' 'No']
Dependents : ['No' 'Yes']
PhoneService : ['No' 'Yes']
MultipleLines : ['No phone service' 'No' 'Yes']
InternetService : ['DSL' 'Fiber optic' 'No']
OnlineSecurity : ['No' 'Yes' 'No internet service']
OnlineBackup : ['Yes' 'No' 'No internet service']
DeviceProtection : ['No' 'Yes' 'No internet service']
TechSupport : ['No' 'Yes' 'No internet service']
StreamingTV : ['No' 'Yes' 'No internet service']
StreamingMovies : ['No' 'Yes' 'No internet service']
Contract : ['Month-to-month' 'One year' 'Two year']
PaperlessBilling : ['Yes' 'No']
PaymentMethod : ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
```

```
       'Credit card (automatic)']
      Churn : ['No' 'Yes']
```

```
data = data.replace(regex=r'No\s[a-z]+\sservice', value='No')
catagorical = [i for i in data.columns if data[i].dtypes == 'object']

for i in catagorical:
    if len(data[i].unique()) == 2:
        data[i] = data[i].map({'Male': 0, 'Female': 1, 'No': 0, 'Yes': 1})
```

```
catagorical = [i for i in data.columns if data[i].dtypes == 'object']

for i in catagorical:
    print(i, ':', data[i].unique())
```

```
     InternetService : ['DSL' 'Fiber optic' 'No']
     Contract : ['Month-to-month' 'One year' 'Two year']
     PaymentMethod : ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
      'Credit card (automatic)']
```

```
data = pd.get_dummies(data)
data.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	O
0	1	0	1	0	1	0	0	
1	0	0	0	0	34	1	0	
2	0	0	0	0	2	1	0	
3	0	0	0	0	45	0	0	
4	1	0	0	0	2	1	0	

5 rows × 27 columns

Fitting the model

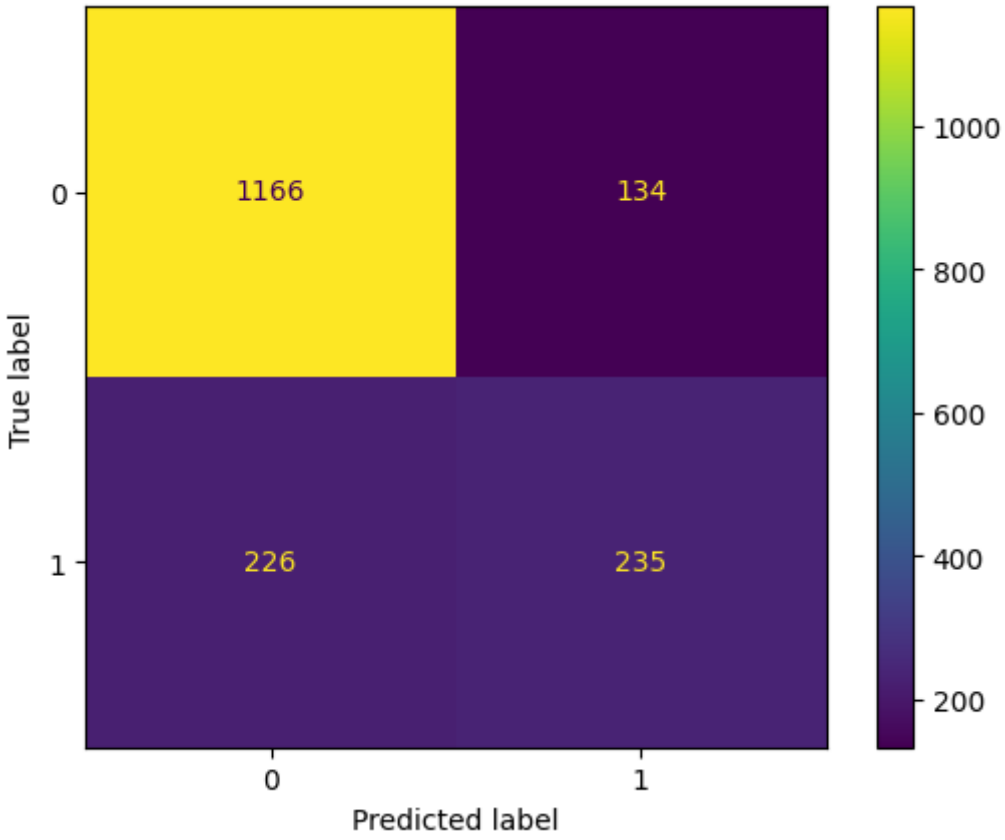
```
y = data.Churn.values
x = data.drop('Churn', axis = 1).values
```

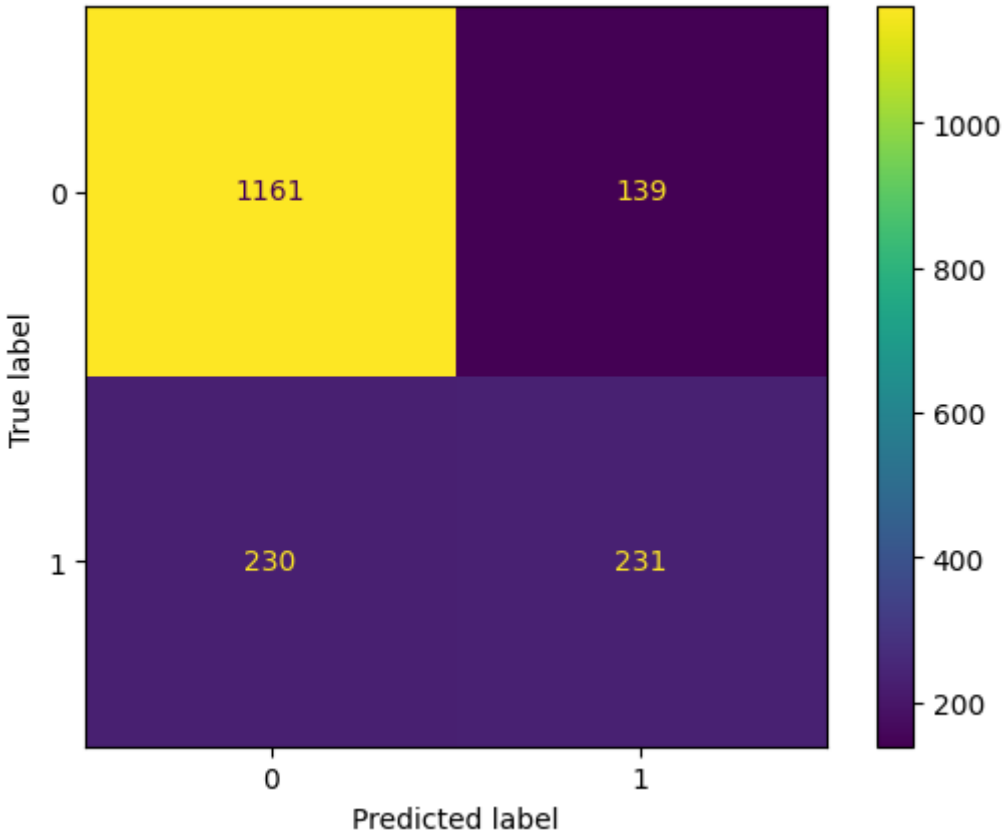
```
x = MinMaxScaler().fit_transform(x)
```

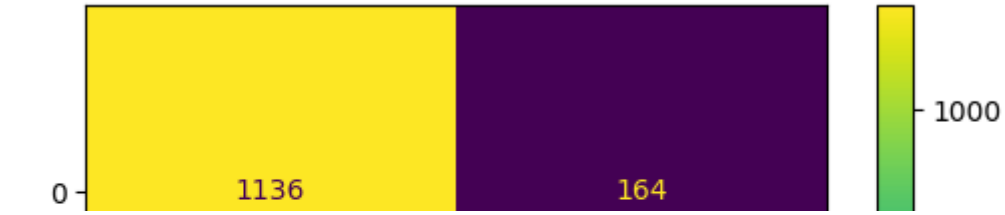
```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=51)
```

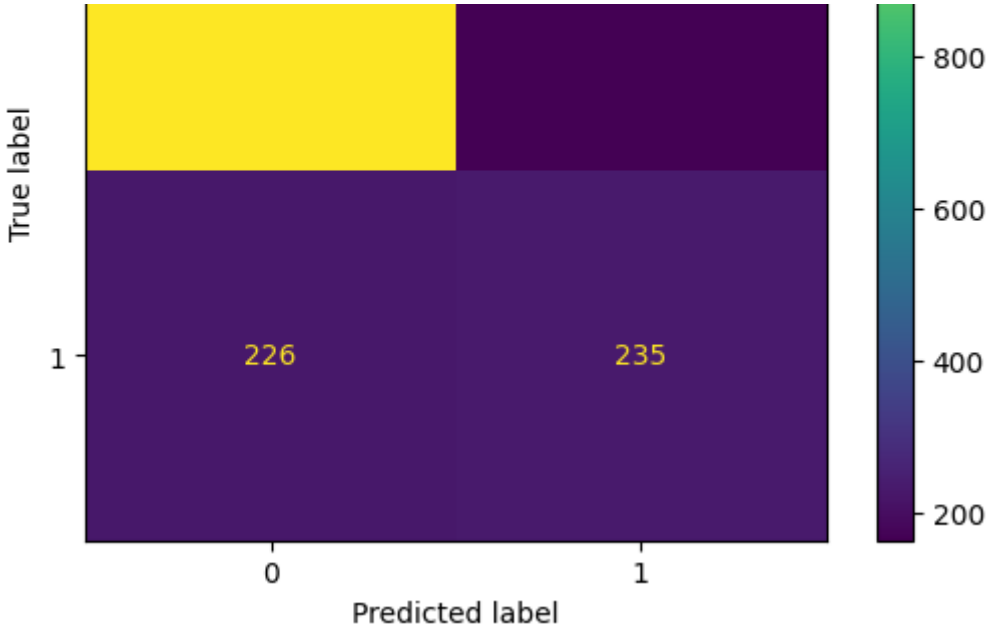
```
rf=RandomForestClassifier()
ab=AdaBoostClassifier()
xg=xgb.XGBClassifier()
sv=SVC(C = 10)


lst=[rf,ab,xg,sv]
for i in lst:
  print('*'*20,i,'*'*20)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print(classification_report(y_test,y_pred))
  print(ConfusionMatrixDisplay.from_predictions(y_test,y_pred))
  print("_"*200)
```


0
1
Predicted label


1166
134
226
235
0
1
True label
0
1
Predicted label
1000
800
600
400
200


1161
139
230
231
0
1
True label
0
1
Predicted label
1000
800
600
400
200


1136
164
0
1000


True label
1
226
235
0
1
Predicted label
800
600
400
200