

AI for business - assignment

Suresh Bojjam, M20AIE313

```
In [113... import pandas as pd
import matplotlib.pyplot as pltxx
```

```
In [114... Data=pd.read_csv("Data_Consumer_ChurnPrediction.csv")
Data
```

```
Out[114]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service
1	5575-GNVDE	Male	0	No	No	34	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service
4	9237-HQITU	Female	0	No	No	2	Yes	No
...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes
7040	4801-JAZL	Female	0	Yes	Yes	11	No	No phone service
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes
7042	3186-AJIEK	Male	0	No	No	66	Yes	No

7043 rows × 21 columns

Data Cleaning

In [115... `Data.drop('customerID',axis=1,inplace = True) #customerID not much useful for predicti`

In [116... `Data['TotalCharges'] = pd.to_numeric(Data['TotalCharges'],errors='coerce') #Convert th`
`Data['TotalCharges'] = Data['TotalCharges'].fillna(Data['TotalCharges'].median()) #Rep`

`boolean_mapping = {'Yes':1, 'No':0}`
`boolean_mapping2 = {'Male':1, 'Female':0}`

`binary_labels = ['Churn', 'Partner', 'Dependents', 'PhoneService', 'PaperlessBilling']`
`for each_label in binary_labels:`
`Data[each_label] = Data[each_label].map(boolean_mapping)`

`Data['gender'] = Data['gender'].map(boolean_mapping2)`

`CatVar = ['MultipleLines','InternetService','OnlineSecurity','OnlineBackup','DevicePr`
`Data = pd.get_dummies(Data, columns = CatVar, drop_first=False)`
`Data`

Out[116]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCh
0	0	0	1	0	1	0	1	
1	1	0	0	0	34	1	0	
2	1	0	0	0	2	1	1	
3	1	0	0	0	45	0	0	
4	0	0	0	0	2	1	1	
...	
7038	1	0	1	1	24	1	1	
7039	0	0	1	1	72	1	1	1
7040	0	0	1	1	11	0	1	
7041	1	1	1	0	4	1	1	
7042	1	0	0	0	66	1	1	1

7043 rows × 41 columns

In [103... `from sklearn.preprocessing import MinMaxScaler`
`sc = MinMaxScaler()`
`Data['tenure'] = sc.fit_transform(Data[['tenure']])`
`Data['MonthlyCharges'] = sc.fit_transform(Data[['MonthlyCharges']])`
`Data['TotalCharges'] = sc.fit_transform(Data[['TotalCharges']])`

In [117... `from sklearn.linear_model import LogisticRegression`
`from sklearn.svm import SVC`
`from sklearn.ensemble import RandomForestClassifier`
`from sklearn.tree import DecisionTreeClassifier`
`from sklearn.naive_bayes import GaussianNB`

```

from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.model_selection import train_test_split

X = Data.drop('Churn', axis=1)
y = Data['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=

def train(alg, name):
    model = alg
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(name)
    print("=====")
    print("accuracy: ", accuracy_score(y_test, y_pred))
    print("precision: ", precision_score(y_test, y_pred))
    print("recall: ", recall_score(y_test, y_pred))
    print("f1_score: ", f1_score(y_test, y_pred, average='weighted'))
    print("\n")
    return model

modeling(LogisticRegression, 'Logistic Regression')
print("")
modeling(SVC, 'SVC Classification')
print("")
modeling(RandomForestClassifier, "Random Forest Classification")
print("")
modeling(DecisionTreeClassifier, "Decision Tree Classification")
print("")
modeling(GaussianNB, "Naive Bayes Classification")
print("")
modeling(MultinomialNB, "MultinomialNB Classification")
print("")
modeling(BernoulliNB, "BernoulliNB Classification")
print("")
modeling(KNeighborsClassifier, "KNeighbors Classification")

```

C:\Users\sbl86126\AppData\Local\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

n_iter_i = _check_optimize_result(

Logistic Regression

```
=====
accuracy: 0.7974443918599148
precision: 0.6257309941520468
recall: 0.5763016157989228
f1_score: 0.7946916817072417
```

SVC Classification

```
=====
accuracy: 0.7363937529578798
precision: 0.0
recall: 0.0
f1_score: 0.6245999888811452
```

```
C:\Users\sb186126\AppData\Local\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

Random Forest Classification

```
=====
accuracy: 0.7898722195929957
precision: 0.6241758241758242
recall: 0.5098743267504489
f1_score: 0.7826170999474519
```

Decision Tree Classification

```
=====
accuracy: 0.7292948414576431
precision: 0.4878836833602585
recall: 0.5421903052064632
f1_score: 0.7336793472790882
```

Naive Bayes Classification

```
=====
accuracy: 0.6786559394226218
precision: 0.4437269372693727
recall: 0.8635547576301615
f1_score: 0.6974991005852047
```

MultinomialNB Classification

```
=====
accuracy: 0.6696639848556555
precision: 0.42828077314343843
recall: 0.755834829443447
f1_score: 0.6891576462226049
```

BernoulliNB Classification

```
=====
accuracy: 0.7094178892569806
precision: 0.47141424272818455
recall: 0.8438061041292639
f1_score: 0.7266304660238987
```

KNeighbors Classification

```
=====
accuracy: 0.7572172266919073
precision: 0.5497737556561086
recall: 0.4362657091561939
f1_score: 0.747569245556604
```

Out[117]:

▼ KNeighborsClassifier

KNeighborsClassifier()