Kingdom of Saudi Arabia
Ministry of Education
University of Jeddah
College of Computer Science and Engineering
Department of Computer Science and Artificial Intelligence

المملكة العربية السعودية
وزارة التعليم
جامعة جدّة
كلية علوم وهندسة الحاسب
قسم علوم الحاسب والذكاء الاصطناعي

# Lab 6
## CCAI 312 Pattern Recognition
### Third Trimester 2023

**Student Name:** Ruba Khalid Alsulami

**Student ID:** 2110618

| | | Max Score | Student Score |
|---|---|---|---|
| PLO S2 / CLO 2 / SO 2 | Task 1 | 4 | |
| Total | | | |

**Task 1: [PLO S2 / CLO 2 / SO 2]**
**[4 marks]**

1. Load the dataset into a pandas dataframe.

```
[1]: import pandas as pd
     wa=pd.read_csv('/kaggle/input/lab6task/WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

2. Perform basic data exploration, including checking for missing values, data types, and summary statistics.

```
# Check the first few rows of the dataset
print(wa.head())
# Check the dimensions of the dataset (number of rows, number of columns)
print(wa.shape)
# Check the data types of each column
print(wa.dtypes)
# Check for missing values in each column
print(wa.isnull().sum())
# Calculate summary statistics for numeric columns
print(wa.describe())
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
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

      MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \
0  No phone service             DSL             No  ...               No
1                No             DSL            Yes  ...              Yes
2                No             DSL            Yes  ...               No
3  No phone service             DSL            Yes  ...              Yes
4                No     Fiber optic             No  ...               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

              PaymentMethod MonthlyCharges  TotalCharges Churn
0          Electronic check          29.85         29.85    No
1              Mailed check          56.95        1889.5    No
2              Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4          Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
(7043, 21)
customerID           object
gender               object
```

```
dtype: object
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
       SeniorCitizen       tenure  MonthlyCharges
count    7043.000000  7043.000000     7043.000000
mean        0.162147    32.371149       64.761692
std         0.368612    24.559481       30.090047
min         0.000000     0.000000       18.250000
25%         0.000000     9.000000       35.500000
50%         0.000000    29.000000       70.350000
75%         0.000000    55.000000       89.850000
max         1.000000    72.000000      118.750000
```

3. Remove customer IDs from the data set

```
wa=wa.drop(['customerID'],axis=1)
wa
```

| | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBack |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | |
| 1 | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | |

4. Convert the predictor variable in a binary numeric variable.

```
wa['Churn'] = wa['Churn'].replace({'Yes': 1, 'No': 0})
wa
```

| | TechSupport | StreamingTV | StreamingMovies | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges | Churn |
|---|---|---|---|---|---|---|---|---|---|
| lo | No | No | No | Month-to-month | Yes | Electronic check | 29.85 | 29.85 | 0 |
| es | No | No | No | One year | No | Mailed check | 56.95 | 1889.5 | 0 |
| lo | No | No | No | Month-to-month | Yes | Mailed check | 53.85 | 108.15 | 1 |
| es | Yes | No | No | One year | No | Bank transfer (automatic) | 42.30 | 1840.75 | 0 |

5. Convert all the categorical variables into dummy variables

```
wa = pd.get_dummies(wa)
```

+ Code + Markdown

6. Show Correlation of "Churn" with other variables

```
correlation = wa.corr()['Churn'].sort_values()
print(correlation)
```

```
tenure           -0.352229
SeniorCitizen     0.150889
MonthlyCharges    0.193356
Churn             1.000000
Name: Churn, dtype: float64
```

7. Apply normalization techniques to standardize the numerical features to have zero mean and unit variance, such as Min-Max scaling, Z-score normalization, or Robust scaling.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaled=pd.DataFrame(scaler.fit_transform(wa.values),columns=wa.columns)
```

+ Code + Markdown

```
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
scaled=pd.DataFrame(scaler.fit_transform(wa.values),columns=wa.columns)
```

8. Split the dataset into training and testing sets using a stratified sampling strategy to preserve the proportion of the target variable in each set.

```
from sklearn.model_selection import train_test_split
X = scaled.drop('Churn', axis=1)
y = scaled['Churn']
X_train, X_test, y_train, y_test = train_test_split(X, y,stratify=y ,test_size=0.2)
```

9. Train the Random Forest model on the training set using the sklearn library.

```
from sklearn.ensemble import RandomForestClassifier
rf= RandomForestClassifier(n_estimators=100,random_state=42)
rf.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)
```

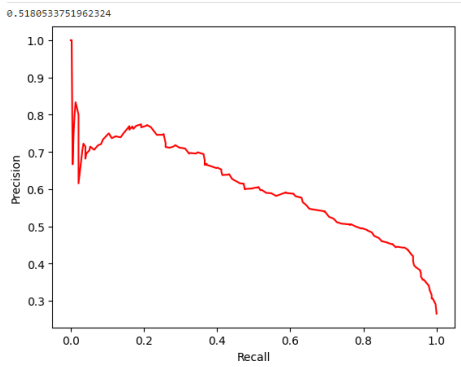
10. Make predictions on the testing set and evaluate the model performance using confusion matrix, precision-recall curve, F1-score, and ROC curve.

```
from sklearn.metrics import confusion_matrix,f1_score,precision_recall_curve,roc_curve,roc_au
import matplotlib.pyplot as plt
# Confusion matrix
y_pred = rf.predict(X_test)
cm=confusion_matrix(y_test, y_pred)
print(cm)
```

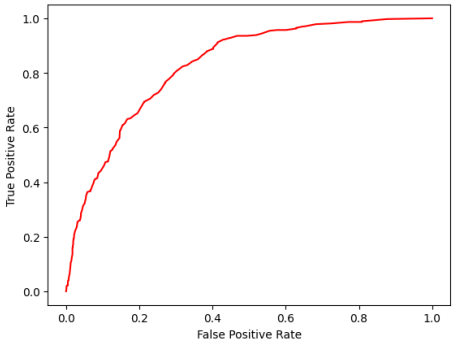
```
[[937  98]
 [209 165]]
```

```
#F1-score
f1=f1_score(y_test,y_pred)
print(f1)
# Precision-recall curve
precision, recall,thresholds = precision_recall_curve(y_test,rf.predict_proba(X_test)[:,1])
plt.plot(recall, precision,'r')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.show()
```

```
0.5180533751962324
```




```
# ROC curve
fpr, tpr, thresholds = roc_curve(y_test,rf.predict_proba(X_test)[:,1])
plt.plot(fpr, tpr,'r')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```




11. Perform 5-fold cross-validation and calculate the mean accuracy score

```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(rf, X, y, cv=5)
print(scores.mean())
```

```
0.7870228522807923
```