

# DATA ANALYTICS WITH COGNOS – GROUP 3

## CUSTOMER CHURN PREDICTION

### PHASE4

### DEVELOPMENT PART 2

#### MODEL BUILDING:

We have completed the data preprocessing in the previous phase, now we build a model that can predict the churning of customers.

#### STEP1: IMPORTING LIBRARIES

We import the libraries necessary for train and test split of data, to encode categorical variables, for feature scaling, to import the model and the metrics to calculate the accuracy and confusion matrix of the model.

```
✓ [22] #importing libraries to design the model and the data to fit the model
Os from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder, StandardScaler
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

We have imported the Random Forest Classifier model from `sklearn.ensemble` as the algorithm we have chosen for our prediction is Random Forest Algorithm. We have also imported the `train_test_split` from `sklearn.model_selection` in order to split the data into training and validation set for the model to learn the training data.

## STEP2: FITTING THE DATA

For the data to fit the model we encode the categorical variables using label encoder, Split the data into training and testing set for the model to learn the training data using `train_test_split`, we perform feature scaling using standard scaler.

```
[23] # to encode categorical variables
label_encoders = {}
for column in X.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    X[column] = label_encoders[column].fit_transform(X[column])

[24] # Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature Scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

## STEP3: MODEL BUILDING AND PREDICTION

We build the RandomForestClassifier model and assign the random\_state as 46 for the consistency of the model and fit the X\_train and y\_train data into the model.

Then we make prediction using the model.predict() function by giving the X\_test as the input.

```
[26] #Building the random forest model and fitting train data
      model = RandomForestClassifier(random_state=46).fit(X_train,y_train)

[27] # Making Prediction
      y_pred = model.predict(X_test)
```

## STEP4: EVALUATION OF MODEL

After the prediction is over we evaluate the data by attaining its accuracy using accuracy\_score(), confusion matrix using confusion\_matrix() and the overall classification report using classification\_report().

```
[28] #attaining the accuracy of the model
      print("Accuracy score",accuracy_score(y_test,y_pred))

      Accuracy score 0.794180269694819

#attaining the confusion matrix
      print("Confusion Matrix:/n",confusion_matrix(y_test,y_pred))

      Confusion Matrix:/n [[944  92]
                           [198 175]]
```

```
[21] #attaining the overall classification report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
No	0.83	0.91	0.87	1036
Yes	0.66	0.47	0.55	373
accuracy			0.79	1409
macro avg	0.74	0.69	0.71	1409
weighted avg	0.78	0.79	0.78	1409

## DATA ANALYSIS AND VISUALIZATION:

### 1) Payment method and churn:

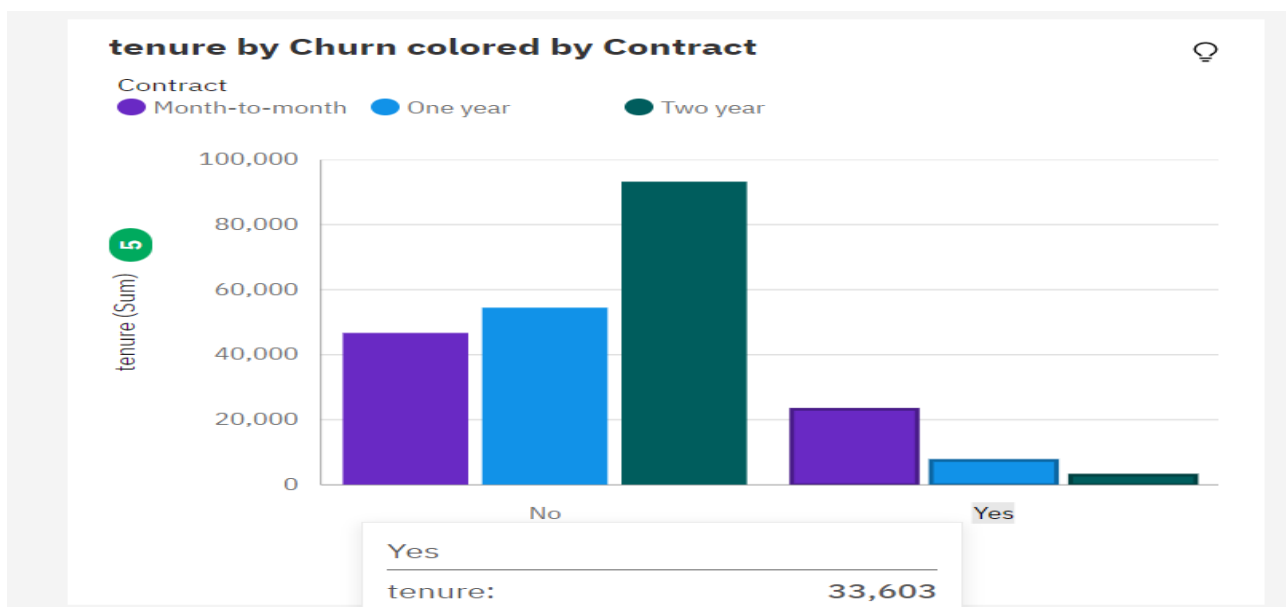
To analyze the effect of payment method on the churners we create a bullet chart to visualize the relationship between them.





It is analyzed that the churners count is higher in the Electronic check payment method than any other payment method.

## 2) TENURE AND CONTRACT WITH RESPECT TO CHURN:



Using the bar plot, We analyze the relationship between churners and their tenure and contract period.

It is clearly observed that the churning decreases when the tenure is high and the contract is of long period like two year.

## **CONCLUSION:**

In this phase we have built the model and had fit the data to the model and had made prediction. We have also made few analyses by visualizing using Cognos.