A

Assesment Report

on

“Classify Customer Churn”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE AIML

By

Prakhar Tiwari(202401100400138)

Under the supervision of

“Mr. Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

The problem focuses on understanding and managing customer behaviour in a telecom company, specifically identifying which customers are likely to leave the service—a phenomenon known as **customer churn**. Telecom companies face significant losses when customers discontinue their services, so being able to predict churn is crucial for customer retention and strategic decision-making.

The goal is twofold. First, to determine **which customers are at risk of churning** by analyzing their usage patterns, service preferences, contract details, and billing information. Recognizing these patterns helps in identifying dissatisfied customers or those who may be tempted by competitors.

Second, the problem involves **segmenting the customer base into different groups** based on similarities in their data. This segmentation is not directly tied to churn but helps understand customer behaviour more broadly. By grouping similar customers together, the business can develop targeted marketing strategies, improve services, and enhance overall customer satisfaction.

Together, churn classification and customer segmentation provide a comprehensive view of the customer base, allowing the telecom company to not only reduce churn but also personalize services and increase profitability. Understanding both aspects is essential for maintaining a competitive edge in a highly dynamic industry.

Methodology

Problem Overview:

A **telecom company** wants to identify:

1. **Which customers are likely to leave (churn)** based on their service usage patterns.
2. **How to group customers into segments** to understand their behaviour better.

Part 1: Classification - Predict Churn

What is churn?

Churn = Yes → The customer has left the company.
Churn = No → The customer is still with the company.

Objective:

Build a **classification model** to **predict churn** based on:

- Customer's demographics (age, gender, senior citizen status)
- Service usage (internet, phone, streaming)
- Billing information (monthly charges, contract type)

Steps Involved:

1. **Data Preprocessing**
   - Handle missing values.
   - Convert string (categorical) data into numeric.
   - Remove irrelevant columns (like customer ID).
2. **Model Building**

    - Use a classifier (like Random Forest) to learn from the data.

3. **Model Evaluation**
    - Predict churn on test data.
    - Generate **confusion matrix** (True Positive, False Negative, etc.).
    - Compute:
        - **Accuracy** – How many correct predictions out of total.
        - **Precision** – Of those predicted to churn, how many actually churned.
        - **Recall** – Of all who actually churned, how many we correctly identified.

Output:

- Confusion matrix heatmap
- Accuracy, Precision, Recall values

Part 2: Clustering - Segment Customers

Objective:

Group customers into **clusters** (segments) based on similar behaviour — without knowing whether they churned.

Steps Involved:

1. **Prepare data** (excluding Churn column)
2. **Normalize** the data (scaling helps clustering)
3. **Apply Clustering Algorithm** (K Means used here)

    - Automatically separates customers into distinct groups (e.g., 4 clusters)

4. **Dimensionality Reduction** (PCA used to reduce features to 2D for plotting)
5. **Visualize Clusters**
    - Scatter plot with clusters labelled.

Output:

- Visual cluster plot (e.g., customers grouped into Cluster 0, 1, 2, 3)

How These Help the Business:

Classification (Churn Prediction):

- Helps **proactively retain** customers at risk of leaving.
- Focus marketing and support efforts on high-risk customers.

Clustering (Segmentation):

- Understand different customer **behavioural groups**.
- Create **personalized offers or services** based on group needs.
- Detect unusual patterns (like high-spending low-usage customers).

Code

```
# Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score, recall_score

from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

# Load dataset
df = pd.read_csv("/content/5. Classify Customer Churn.csv")
```

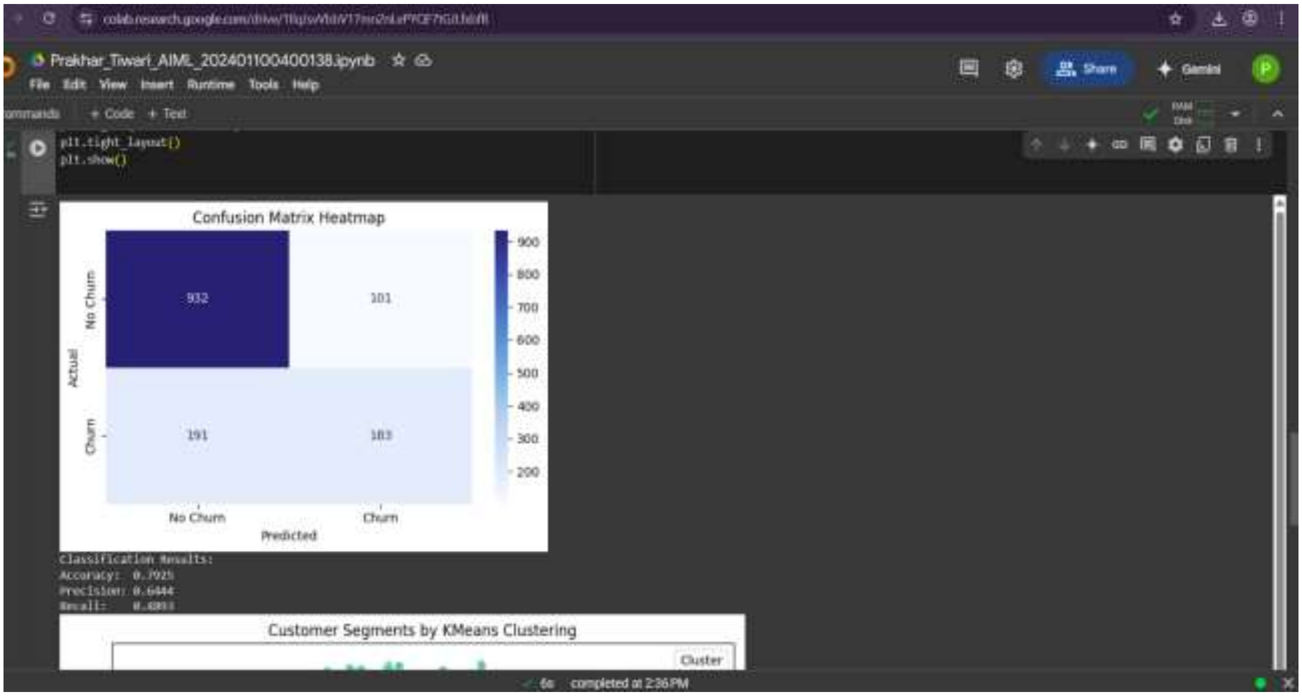
```
### ----------------------------

### PART 1: CLASSIFICATION

### ----------------------------


# Convert TotalCharges to numeric and handle missing values

df['TotalCharges'] = pd.to_numeric(df['TotalCharges'],
errors='coerce')

df.dropna(inplace=True)


# Drop customerID

df.drop(columns=['customerID'], inplace=True)


# Encode target variable

df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})


# Encode categorical features

le = LabelEncoder()

categorical_cols = df.select_dtypes(include=['object']).columns

df[categorical_cols] = df[categorical_cols].apply(le.fit_transform)
```

```
# Features and target
X = df.drop('Churn', axis=1)
y = df['Churn']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train Random Forest Classifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
```

```
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['No Churn', 'Churn'], yticklabels=['No Churn', 'Churn'])
plt.title("Confusion Matrix Heatmap")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.show()

# Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print("Classification Results:")
print(f"Accuracy:  {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall:    {recall:.4f}")

### ----------------------------
```

```
### PART 2: CLUSTERING
### ----------------------------

# For clustering, remove target column
features = df.drop(columns=['Churn'])

# Standardize data
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# Dimensionality reduction for visualization
pca = PCA(n_components=2)
pca_features = pca.fit_transform(scaled_features)

# Apply KMeans clustering
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(scaled_features)

# Add cluster and PCA info to dataframe
```

```
df['Cluster'] = clusters
df['PCA1'] = pca_features[:, 0]
df['PCA2'] = pca_features[:, 1]

# Plot clusters
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='PCA1', y='PCA2', hue='Cluster', palette='Set2', s=60)
plt.title("Customer Segments by KMeans Clustering")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend(title="Cluster")
plt.tight_layout()
plt.show()
```



Prakhar_Tiwari_AIML_202401100400138.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Share
Gemini
plt.tight_layout()
plt.show()
Confusion Matrix Heatmap
No Churn
Churn
Actual
Predicted
932
101
191
183
900
800
700
600
500
400
300
200
Classification Results:
Accuracy: 0.7925
Precision: 0.6444
Recall:
Customer Segments by KMeans Clustering
Cluster
completed at 2:36 PM

Prakhar_Tiwari_AIML_202401100400138.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Share
Gemini
Customer Segments by KMeans Clustering
Cluster
0
1
2
3
Principal Component 2
Principal Component 1
−4
−2
0
2
4
6
3
2
1
0
−1
−2
−3
completed at 2:36 PM