# Day 9 – Random Forest, Unsupervised Learning & Real-World Applications

## 📘 Today’s Highlights

Today’s session included practical and theoretical concepts covering:

1. ✅ **Customer Churn Prediction using Random Forest**
2. ✅ **Introduction to Unsupervised Learning**
3. ✅ **Heart Disease Prediction – Conceptual Overview**

We saw how tree-based models can be used to solve real-world classification problems using cleaned and preprocessed data.

## 🌲 Random Forest Recap

- Random Forest is an **ensemble learning** algorithm made of multiple decision trees.
- It improves accuracy by averaging predictions and reducing overfitting.
- It is effective on both categorical and numerical data, often used in real-world classification tasks like **customer churn** and **disease diagnosis**.

## 📊 Customer Churn Prediction with Random Forest

### ✅ Dataset Used: `Telco Customer Churn Dataset`

This dataset includes customer info like tenure, contract type, payment method, and whether they left the company (`Churn`).

### □ Preprocessing & Encoding:

- Removed rows with `tenure = 0`
- Mapped `SeniorCitizen` from numeric to `"Yes"`/`"No"`
- Applied **Label Encoding** on categorical features

### □ Python Code:

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

from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
# Load dataset
df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')

# Remove invalid tenure rows
df.drop(labels=df[df['tenure'] == 0].index, axis=0, inplace=True)

# Map SeniorCitizen to Yes/No
df["SeniorCitizen"] = df["SeniorCitizen"].map({0: "No", 1: "Yes"})

# Encode all object columns
def object_to_int(series):
    if series.dtype == 'object':
        return LabelEncoder().fit_transform(series)
    return series

df = df.apply(lambda x: object_to_int(x))

# Split data
X = df.drop(columns=['Churn'])
y = df['Churn'].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=40, stratify=y)

# Train Random Forest
model_rf = RandomForestClassifier(
    n_estimators=500, oob_score=True, n_jobs=-1,
    random_state=50, max_features="sqrt", max_leaf_nodes=30
)
model_rf.fit(X_train, y_train)

# Evaluate
prediction_test = model_rf.predict(X_test)
print("Accuracy:", accuracy_score(y_test, prediction_test))

# Confusion Matrix
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, prediction_test), annot=True,
fmt="d", cmap="coolwarm")
plt.title("Customer Churn Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

### ✓ Sample Output:

```
Accuracy: 0.80  (or ~80%)
```

🔍 The confusion matrix showed a balanced prediction between churn and non-churn classes.

### □ What is Unsupervised Learning?

- In **unsupervised learning**, the algorithm tries to find **patterns** in data without any labeled output.
- Common types:
  - **Clustering** (e.g., customer segmentation)
  - **Dimensionality reduction** (e.g., PCA)
- Unlike supervised learning, there's **no target column** — the model groups data based on similarity.

### 📌 Real-world examples:

- Grouping users by browsing habits
- Market basket analysis in e-commerce
- News article categorization

We’ll explore clustering (like **K-Means**) in upcoming sessions.

---

### ❤□ Heart Disease Prediction – Concept Explained

Heart Disease Prediction is a **real-world application** of supervised classification algorithms.

- The model uses medical parameters like age, blood pressure, cholesterol, heart rate, etc.
- Algorithms like **Logistic Regression**, **SVM**, or **Random Forest** can be used.
- Goal: Predict whether a person is at **risk of heart disease** (`1`) or not (`0`)

Even though we didn’t implement it today, this concept was introduced to show how **ML can support healthcare diagnostics**.

---

### 📝 Conclusion

- Built and evaluated a real-world **Customer Churn model** using Random Forest.
- Learned theory of **Unsupervised Learning** and its applications.
- Discussed use of ML in **healthcare**, like **heart disease prediction**.
- Understood importance of **data preprocessing**, **label encoding**, and **model evaluation**.

This session was highly practical and expanded our view of real ML applications beyond academic datasets.