Lesson Plan: Classification Techniques in Data Mining

⏰ Duration: 60 minutes

🎯 Target Audience: Undergraduate / Beginner-level learners

🧩 Prerequisites: Basic understanding of data mining, supervised learning, and Python.

Lesson Objectives (SMART)

By the end of this lesson, students will be able to:
- Define classification in the context of data mining.
- Identify and describe common classification techniques.
- Compare the strengths and limitations of each technique.
- Demonstrate a basic classification technique using Python.
- Evaluate classifier performance using accuracy and confusion matrix.

Lesson Structure (60 minutes)

Time	Activity	Description
0–5 mins	Engage / Warm-Up	Ask: “How does your email client know what is spam?” → Introduce classification.
5–15 mins	Introduction	Define classification. Explain class labels, features, training, and testing.
15–30 mins	Technique Overview	Overview of techniques: - Decision Trees - Naive Bayes - k-NN - Support Vector Machines (SVM) - Neural Networks
30–45 mins	Hands-On Coding	Use a dataset (e.g., Titanic or Iris) in Python:

		Train and test classifiers (e.g., Decision Tree or Naive Bayes) using scikit-learn.
45–55 mins	Evaluation Methods	Discuss accuracy, confusion matrix, precision, recall, F1-score. Show evaluation output from model.
55–60 mins	Wrap-Up / Q&A	Recap key points, answer questions, share further reading and practice exercises.

Materials & Resources

- Laptop with Jupyter Notebook or Google Colab
- Python (pandas, scikit-learn, matplotlib)
- Dataset: Titanic or Iris
- Slides or handouts

Sample Code Snippet

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Load data
data = load_iris()
X = data.data
y = data.target

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# Train model
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Predict
y_pred = clf.predict(X_test)
```

```
# Evaluate
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Sample Quiz / Assessment

1. Name two advantages of using Decision Trees.
2. What assumption does Naive Bayes make?
3. Which classifier is distance-based?
4. Interpret this confusion matrix output.
5. What metrics are used to evaluate classification performance?

Further Reading / Homework

- scikit-learn documentation on classification:
https://scikit-learn.org/stable/supervised_learning.html
- “Data Mining: Concepts and Techniques” by Han, Kamber & Pei
- Practice: Try building SVM and Naive Bayes classifiers on the Titanic dataset