Assignment 4: Text Classification with SVM and TF-IDF (Do it in Week 3)

Task:

* Load a text classification dataset.
* Preprocess the text data using TF-IDF vectorization.
* Split the dataset into training and testing sets using train\_test\_split.
* Train a Support Vector Machine (SVM) model on the TF-IDF transformed data.
* Evaluate the model using accuracy and F1-score.



















# Import necessary libraries

from sklearn.datasets import fetch\_20newsgroups

from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, f1\_score

# Load dataset (You can use any other text classification dataset)

newsgroups = fetch\_20newsgroups(subset='all', shuffle=True, random\_state=42)

















# Preprocess the text data using TF-IDF vectorization

tfidf\_vectorizer = TfidfVectorizer(stop\_words='english')

X = tfidf\_vectorizer.fit\_transform(newsgroups.data) y = newsgroups.target

# Split the dataset 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)

































# Train a Support Vector Machine (SVM) model

svm\_model = SVC(kernel='linear', random\_state=42) svm\_model.fit(X\_train, y\_train)

# Predict on the testing set

y\_pred = svm\_model.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred, average='weighted')

# Print evaluation metrics

print("Accuracy:", accuracy) print("F1 Score:", f1)

Output:

Accuracy: 0.9180371352785146

F1 Score: 0.9186626638965073

