2024

ML ASSIGNMENT 4

ROHA ARSLAN (FA21-BSE-116)

DR. MUHAMMAD SHARJEEL

COMSATS University Islamabad, Lahore Campus

Course Title:	Machine Learning				Course Code:	CSC354	Credit Hours:	3(3,0)
Resource Person:	Dr. Muhammad Sharjeel				Programme Name:	BSSE		
Semester:	6th	Batch:	FA21	Section:	A, B	Max Marks:	10	

Assignment 4: **Due Date/Time: Wednesday, 29th May, 23:59**

Submission:
Upload the assignment solution PDF file on CUOnline.

Note: Dataset required is available in the Google Drive shared folder.

Question1: [CLO-5] - [Bloom Taxonomy Level: <Creating>]

Your task in this assignment is to use a number of ML algorithms to classify real vs. fake news. The dataset contains 1298 "fake" and 1968 "real" news instances, stored in two separate text files. In each file, one news text appears on a line single with words separated by spaces, so just use Python's str.split() to split text into words.

Your first task is to load the data, preprocess it (you are free to use any text preprocessing pipeline), and then generate feature vectors using a vectorizer (either use a simple tf.idf or more advanced text embeddings vectors). Next, train Naïve Bayes, Random Forest, and Support Vectors Machines models on the entire dataset using the following guidelines;

- For Naïve Bayes, use simple, multinomial, and bernoulli implementation
- For Random Forest, choose the best parameters using a grid search
- For Support Vectors Machines use linear and RBF kernels

For each algorithm settings,

- Use train/test split of 70/30 with random and stratified splitting.
- Use 10-fold cross validation with random and stratified distribution.

GITHUB LINK : Assignment4

CODE

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score,
GridSearchCV, StratifiedKFold
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Load the datasets
with open('/content/fake.txt') as f:
    fake_news = f.readlines()

with open('/content/real.txt') as f:
    real_news = f.readlines()

# Create DataFrame
data_fake = pd.DataFrame(fake_news, columns=['text'])
data_fake['label'] = 0
data_real = pd.DataFrame(real_news, columns=['text'])
data_real['label'] = 1

data = pd.concat([data_fake, data_real], ignore_index=True)

# Preprocess the text data
def preprocess(text):

    return text.strip()

data['text'] = data['text'].apply(preprocess)

# Generate TF-IDF features
vectorizer = TfidfVectorizer(max_features=5000)
X = vectorizer.fit_transform(data['text']).toarray()
y = data['label']

# Split the data (random and stratified)
X_train_rand, X_test_rand, y_train_rand, y_test_rand =
train_test_split(X, y, test_size=0.3, random_state=42)
X_train_strat, X_test_strat, y_train_strat, y_test_strat =
train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

# Define the models
models = {
```

```
    'GaussianNB': GaussianNB(),
    'MultinomialNB': MultinomialNB(),
    'BernoulliNB': BernoulliNB(),
    'RandomForest': RandomForestClassifier(),
    'SVM_linear': SVC(kernel='linear'),
    'SVM_rbf': SVC(kernel='rbf')
}

# Function to train and evaluate models
def evaluate_models(models, X_train, X_test, y_train, y_test):
    results = {}
    for name, model in models.items():
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        results[name] = accuracy
        print(f'{name} Accuracy: {accuracy}')
        print(classification_report(y_test, y_pred))
    return results

print("Random Split Evaluation:")
evaluate_models(models, X_train_rand, X_test_rand, y_train_rand,
y_test_rand)

print("\nStratified Split Evaluation:")
evaluate_models(models, X_train_strat, X_test_strat, y_train_strat,
y_test_strat)

# Grid Search for Random Forest
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30]
}
grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=10,
scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train_strat, y_train_strat)
print(f'Best parameters for Random Forest: {grid_search.best_params_}')

# 10-fold cross-validation (random and stratified)
def cross_validate_models(models, X, y):
    for name, model in models.items():
        # Random 10-fold CV
        scores_random = cross_val_score(model, X, y, cv=10)
        print(f'{name} 10-fold CV Random Accuracy:
{np.mean(scores_random)}')

        # Stratified 10-fold CV
        skf = StratifiedKFold(n_splits=10)
```

```
        scores_stratified = cross_val_score(model, X, y, cv=skf)
        print(f'{name} 10-fold CV Stratified Accuracy:
{np.mean(scores_stratified)}')

print("\nCross-Validation Evaluation:")
cross_validate_models(models, X, y)
```

RESULTS

```
macro avg       0.84      0.78      0.80       980
weighted avg       0.83      0.82      0.81       980

BernoulliNB Accuracy: 0.8275510204081633
              precision    recall  f1-score   support

           0       0.91      0.62      0.74       389
           1       0.80      0.96      0.87       591

    accuracy                           0.83       980
   macro avg       0.85      0.79      0.81       980
weighted avg       0.84      0.83      0.82       980

RandomForest Accuracy: 0.8591836734693877
              precision    recall  f1-score   support

           0       0.90      0.73      0.80       389
           1       0.84      0.95      0.89       591

    accuracy                           0.86       980
   macro avg       0.87      0.84      0.85       980
weighted avg       0.86      0.86      0.86       980

SVM_linear Accuracy: 0.8428571428571429
              precision    recall  f1-score   support

           0       0.83      0.75      0.79       389
           1       0.85      0.90      0.87       591

    accuracy                           0.84       980
   macro avg       0.84      0.83      0.83       980
weighted avg       0.84      0.84      0.84       980

SVM_rbf Accuracy: 0.8346938775510204
              precision    recall  f1-score   support

           0       0.86      0.69      0.77       389
           1       0.82      0.93      0.87       591

    accuracy                           0.83       980
   macro avg       0.84      0.81      0.82       980
weighted avg       0.84      0.83      0.83       980
```

```
Best parameters for Random Forest: {'max_depth': None, 'n_estimators': 50}
```

```
Cross-Validation Evaluation:
GaussianNB 10-fold CV Random Accuracy: 0.7657520496801186
GaussianNB 10-fold CV Stratified Accuracy: 0.7657520496801186
MultinomialNB 10-fold CV Random Accuracy: 0.8236355790698111
MultinomialNB 10-fold CV Stratified Accuracy: 0.8236355790698111
BernoulliNB 10-fold CV Random Accuracy: 0.8545505712838409
BernoulliNB 10-fold CV Stratified Accuracy: 0.8545505712838409
RandomForest 10-fold CV Random Accuracy: 0.8570073732200146
RandomForest 10-fold CV Stratified Accuracy: 0.8542494512298081
SVM_linear 10-fold CV Random Accuracy: 0.843559220277293
SVM_linear 10-fold CV Stratified Accuracy: 0.843559220277293
SVM_rbf 10-fold CV Random Accuracy: 0.8441727172098086
SVM_rbf 10-fold CV Stratified Accuracy: 0.8441727172098086
```

REPORT

BACKGROUND :

This study aims to classify news articles as real or fake using machine learning algorithms, given the growing concern about misinformation. The dataset used includes text data from real and fake news sources, processed for effective model training and evaluation. The dataset includes two files: real.txt and fake.txt. The text files were read and combined into a single DataFrame, labeled as 1 and 0, respectively. The TfidfVectorizer was used to transform the text data into TF-IDF vectors for model training.

The experimental setup employed a combination of Naive Bayes (GaussianNB, MultinomialNB, BernoulliNB), Random Forest Classifier, and Support Vector Machine (SVM). Validation techniques included training/test splits of 70% training and 30% testing, a stratified split of 70% training and 30% testing while maintaining class distribution, and 10-fold cross-validation. The study also employed random distribution without considering class distribution and stratified distribution with stratified class distribution.

RESULT

Train/Test Split

Random Split

- **GaussianNB**
    - Accuracy: 0.736
    - Precision, Recall, F1-Score:
        - Fake: 0.67, 0.73, 0.70
        - Real: 0.79, 0.74, 0.76

- **MultinomialNB**
    - Accuracy: 0.792
    - Precision, Recall, F1-Score:
        - Fake: 0.89, 0.58, 0.70
        - Real: 0.75, 0.95, 0.84
- **BernoulliNB**
    - Accuracy: 0.835
    - Precision, Recall, F1-Score:
        - Fake: 0.94, 0.65, 0.77
        - Real: 0.79, 0.97, 0.87
- **RandomForest**
    - Accuracy: 0.847
    - Precision, Recall, F1-Score:
        - Fake: 0.90, 0.72, 0.80
        - Real: 0.82, 0.94, 0.88
- **SVM (Linear)**
    - Accuracy: 0.837
    - Precision, Recall, F1-Score:
        - Fake: 0.85, 0.75, 0.80
        - Real: 0.83, 0.90, 0.86
- **SVM (RBF)**
    - Accuracy: 0.822
    - Precision, Recall, F1-Score:
        - Fake: 0.87, 0.68, 0.76
        - Real: 0.80, 0.92, 0.86

Stratified Split

- **GaussianNB**
    - Accuracy: 0.734
    - Precision, Recall, F1-Score:

        - Fake: 0.66, 0.68, 0.67
        - Real: 0.79, 0.77, 0.78
- **MultinomialNB**
    - Accuracy: 0.818
    - Precision, Recall, F1-Score:
        - Fake: 0.89, 0.62, 0.73
        - Real: 0.79, 0.95, 0.86
- **BernoulliNB**
    - Accuracy: 0.828
    - Precision, Recall, F1-Score:
        - Fake: 0.91, 0.62, 0.74
        - Real: 0.80, 0.96, 0.87
- **RandomForest**
    - Accuracy: 0.859
    - Precision, Recall, F1-Score:
        - Fake: 0.90, 0.73, 0.80
        - Real: 0.84, 0.95, 0.89
- **SVM (Linear)**
    - Accuracy: 0.843
    - Precision, Recall, F1-Score:
        - Fake: 0.83, 0.75, 0.79
        - Real: 0.85, 0.90, 0.87
- **SVM (RBF)**
    - Accuracy: 0.835
    - Precision, Recall, F1-Score:
        - Fake: 0.86, 0.69, 0.77
        - Real: 0.82, 0.93, 0.87

10-Fold Cross-Validation

Random Distribution

- GaussianNB: Accuracy: 0.766
- MultinomialNB: Accuracy: 0.824
- BernoulliNB: Accuracy: 0.855
- RandomForest: Accuracy: 0.857
- SVM (Linear): Accuracy: 0.844
- SVM (RBF): Accuracy: 0.844

Stratified Distribution

- GaussianNB: Accuracy: 0.766
- MultinomialNB: Accuracy: 0.824
- BernoulliNB: Accuracy: 0.855
- RandomForest: Accuracy: 0.854
- SVM (Linear): Accuracy: 0.844
- SVM (RBF): Accuracy: 0.844

CONCLUSION

The Random Forest classifier performed best in random and stratified splits, as well as cross-validation. Naive Bayes models, particularly BernoulliNB, also showed strong performance. Stratified validation techniques yielded more balanced and reliable performance metrics, highlighting the importance of maintaining class distribution during validation. It took 32 minutes to execute and train.