Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регресии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода train_test_split разделите выборку на обучающую и тестовую.
4. Обучите следующие ансамблевые модели:

- две модели группы бэггинга (бэггинг или случайный лес или сверхслучайные деревья);
- AdaBoost;
- градиентный бустинг.

1. Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.

Описание датасета: Fake News Detection

Цель: задача бинарной классификации — определить, является ли новостная статья фейковой или настоящей.

Структура данных:

- **title** — заголовок статьи (краткое описание)
- **text** — полный текст статьи
- **date** — дата публикации
- **source** — источник (например, BBC, CNN). Есть пропущенные значения (~5%)
- **author** — имя автора. Есть пропущенные значения (~5%)
- **category** — рубрика статьи (Политика, Спорт и т.д.)
- **label** — целевая переменная: `real` или `fake`

Особенности:

- ~5% пропущенных значений в `source` и `author`
- Реалистичное распределение меток
- Большой текстовый признак (`text`) для обработки NLP

Импорт необходимых библиотек

```
# Работа с данными
import pandas as pd
import numpy as np

# Визуализация
import matplotlib.pyplot as plt
import seaborn as sns

# Предобработка текста и признаков
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer

# Модели ансамблей
from sklearn.ensemble import (
    BaggingClassifier,
    RandomForestClassifier,
    ExtraTreesClassifier,
    AdaBoostClassifier,
    GradientBoostingClassifier
)

# Модели для базовых классификаторов
from sklearn.tree import DecisionTreeClassifier

# Метрики
from sklearn.metrics import classification_report, accuracy_score
```

Загрузка и первичный анализ данных

```
df = pd.read_csv("fake_news_dataset.csv")

print("Размер датасета:", df.shape)

display(df.head())

print("\nКоличество пропущенных значений по столбцам:")
print(df.isnull().sum())

print("\nРаспределение меток:")
print(df['label'].value_counts())
```

```
Размер датасета: (20000, 7)

                                  title  \
0               Foreign Democrat final.
1   To offer down resource great point.
2          Himself church myself carry.
3                  You unit its should.
4  Billion believe employee summer how.

                                                text        date
source  \
0  more tax development both store agreement lawy...  2023-03-10  NY
Times
1  probably guess western behind likely next inve...  2022-05-25  Fox
News
2  them identify forward present success risk sev...  2022-09-01
CNN
3  phone which item yard Republican safe where po...  2023-02-07
```

```
Reuters
4  wonder myself fact difficult course forget exa...  2023-04-03
CNN

                  author    category label
0          Paula George    Politics  real
1           Joseph Hill    Politics  fake
2        Julia Robinson    Business  fake
3  Mr. David Foster DDS     Science  fake
4         Austin Walker  Technology  fake


Количество пропущенных значений по столбцам:
title          0
text           0
date           0
source      1000
author      1000
category       0
label          0
dtype: int64

Распределение меток:
label
fake    10056
real     9944
Name: count, dtype: int64
```

```
df['author'] = df['author'].fillna("unknown")
df['source'] = df['source'].fillna("unknown")
```

Формирование текстового признака для векторизации

```
df['combined_text'] = df['title'] + ' ' + df['text'] + ' ' +
df['source'] + ' ' + df['category']

print("Пример объединённого текста:")
print(df['combined_text'].iloc[0])
```

```
Пример объединённого текста:
Foreign Democrat final. more tax development both store agreement
lawyer hear outside continue reach difference yeah figure your power
fear identify there protect security great national nothing fast story
why late nearly bit cost tough since question to power almost future
young conference behind ahead building teach million box receive Mrs
risk benefit month compare environment class imagine you vote
community reason set once idea him answer many how purpose deep
training game own true language garden of partner result face military
```

```
discover discover data glass bed maintain test way development across
top culture glass yes decision hope necessary as trade organization
talk debate peace stay community development six wide write itself
several fight teach billion for common fear we personal church
establish store kind hundred debate hotel cut sister audience sound
case that stay within information trouble be debate great themselves
responsibility force people hundred bar miss others sometimes build
room interesting however charge what especially north no especially us
travel industry about including face ten behind black series place age
soldier early trouble middle would along case what money significant
sound song reason poor free want thank cultural range shoulder rest
movie political fear hear past leader up edge professor determine law
act change middle prove say notice travel open director argue economic
seven game matter season NY Times Politics
```

Очистка текста и удаление ненужных признаков

```python
import re

df = df.drop(columns=['author', 'date'])
def clean_text(text):
    text = str(text).lower()                                # в нижний
регистр
    text = re.sub(r'\W', ' ', text)                         # убираем
спецсимволы
    text = re.sub(r'\d+', '', text)                         # удаляем
цифры
    text = re.sub(r'\s+', ' ', text).strip()                # убираем
лишние пробелы
    text = re.sub(r'(.)\1{3,}', r'\1', text)                # удаляем
повторяющиеся символы (например, "soooo" → "so")
    return text
df['combined_text'] = df['combined_text'].apply(clean_text)

print("Пример очищенного текста:")
print(df['combined_text'].iloc[0])
```

```
Пример очищенного текста:
foreign democrat final more tax development both store agreement
lawyer hear outside continue reach difference yeah figure your power
fear identify there protect security great national nothing fast story
why late nearly bit cost tough since question to power almost future
young conference behind ahead building teach million box receive mrs
risk benefit month compare environment class imagine you vote
community reason set once idea him answer many how purpose deep
training game own true language garden of partner result face military
discover discover data glass bed maintain test way development across
top culture glass yes decision hope necessary as trade organization
talk debate peace stay community development six wide write itself
```

```
several fight teach billion for common fear we personal church
establish store kind hundred debate hotel cut sister audience sound
case that stay within information trouble be debate great themselves
responsibility force people hundred bar miss others sometimes build
room interesting however charge what especially north no especially us
travel industry about including face ten behind black series place age
soldier early trouble middle would along case what money significant
sound song reason poor free want thank cultural range shoulder rest
movie political fear hear past leader up edge professor determine law
act change middle prove say notice travel open director argue economic
seven game matter season ny times politics
```

```
df.head(10)
```

```
                                              title  \
0                           Foreign Democrat final.
1               To offer down resource great point.
2                      Himself church myself carry.
3                              You unit its should.
4              Billion believe employee summer how.
5  Method purpose mission approach professor short.
6                                Laugh member step.
7                   Center measure difference dark.
8        Moment make those affect first difference.
9           Reason physical contain total decision.

                                                text      source  \
0  more tax development both store agreement lawy...    NY Times
1  probably guess western behind likely next inve...    Fox News
2  them identify forward present success risk sev...         CNN
3  phone which item yard Republican safe where po...     Reuters
4  wonder myself fact difficult course forget exa...         CNN
5  affect too bill whether kind project turn offi...     Reuters
6  often along newspaper establish fall president...         CNN
7  ready movement bed increase during or history ...    NY Times
8  officer mention dream fill later foot suffer d...    Fox News
9  choose anything treat beyond political minute ...  Daily News

        category label
combined_text
0       Politics  real  foreign democrat final more tax development
bo...
1       Politics  fake  to offer down resource great point probably
gu...
2       Business  fake  himself church myself carry them identify
forw...
3        Science  fake  you unit its should phone which item yard
repu...
4     Technology  fake  billion believe employee summer how wonder
mys...
```

```
5         Health  real  method purpose mission approach professor
shor...
6       Business  fake  laugh member step often along newspaper
establ...
7         Sports  fake  center measure difference dark ready
movement ...
8  Entertainment  fake  moment make those affect first difference
offi...
9         Health  real  reason physical contain total decision
choose ...
```

TF-IDF векторизация текста и кодирование целевой переменной

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

tfidf = TfidfVectorizer(max_features=5000, stop_words='english')
X = tfidf.fit_transform(df['combined_text'])

le = LabelEncoder()
y = le.fit_transform(df['label'])

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print("Размер обучающей выборки:", X_train.shape)
print("Размер тестовой выборки:", X_test.shape)
```

```
Размер обучающей выборки: (16000, 775)
Размер тестовой выборки: (4000, 775)
```

Обучение ансамблевых моделей и сравнение качества

На этом этапе обучаются 5 ансамблевых моделей:

- **Бэггинг:** BaggingClassifier, RandomForestClassifier
- **Сверхслучайные деревья:** ExtraTreesClassifier
- **Boosting:** AdaBoostClassifier, GradientBoostingClassifier

Оценивается качество каждой модели по метрике Accuracy.

```
from sklearn.ensemble import (
    BaggingClassifier,
    RandomForestClassifier,
    ExtraTreesClassifier,
```

```
    AdaBoostClassifier,
    GradientBoostingClassifier
)
from sklearn.metrics import accuracy_score

models = {
    "Bagging (DecisionTree)": BaggingClassifier(n_estimators=100,
random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100,
random_state=42),
    "Extra Trees": ExtraTreesClassifier(n_estimators=100,
random_state=42),
    "AdaBoost": AdaBoostClassifier(n_estimators=100, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(n_estimators=100,
random_state=42)
}
results = {}

for name, model in models.items():
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    acc = accuracy_score(y_test, preds)
    results[name] = acc
    print(f"{name}: Accuracy = {acc:.4f}")

results_df = pd.DataFrame.from_dict(results, orient='index',
columns=['Accuracy']).sort_values(by='Accuracy', ascending=False)
display(results_df)
```

```
Bagging (DecisionTree): Accuracy = 0.5082
Random Forest: Accuracy = 0.4988
Extra Trees: Accuracy = 0.5085
AdaBoost: Accuracy = 0.5120
Gradient Boosting: Accuracy = 0.5060
```

```
                        Accuracy
AdaBoost                 0.51200
Extra Trees              0.50850
Bagging (DecisionTree)   0.50825
Gradient Boosting        0.50600
Random Forest            0.49875
```