湖北大学
HUBEI UNIVERSITY

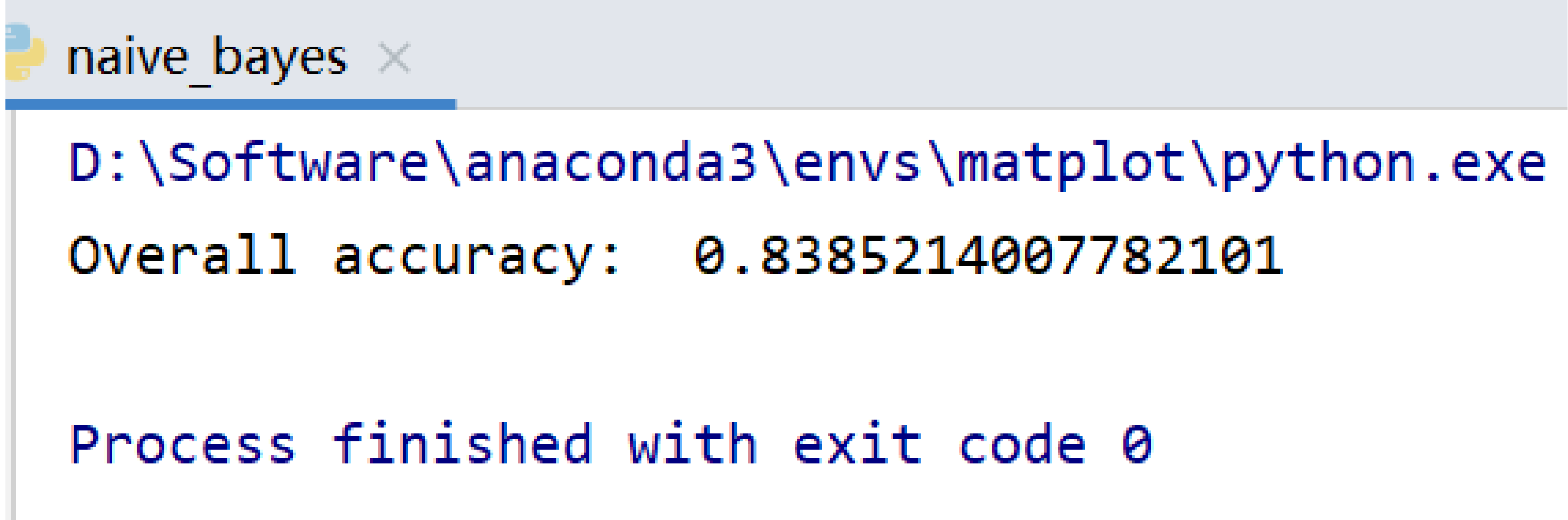
实验三　朴素贝叶斯分类器

本周实验内容

- 安装机器学习库scikit-learn
- 学会调用sklearn中的朴素贝叶斯分类器，对20newsgroups数据集进行分类
- 完成并提交实验报告（4月9日上课前交给班长）

实验要求

- **结果示例：**分类准确率达到80%以上

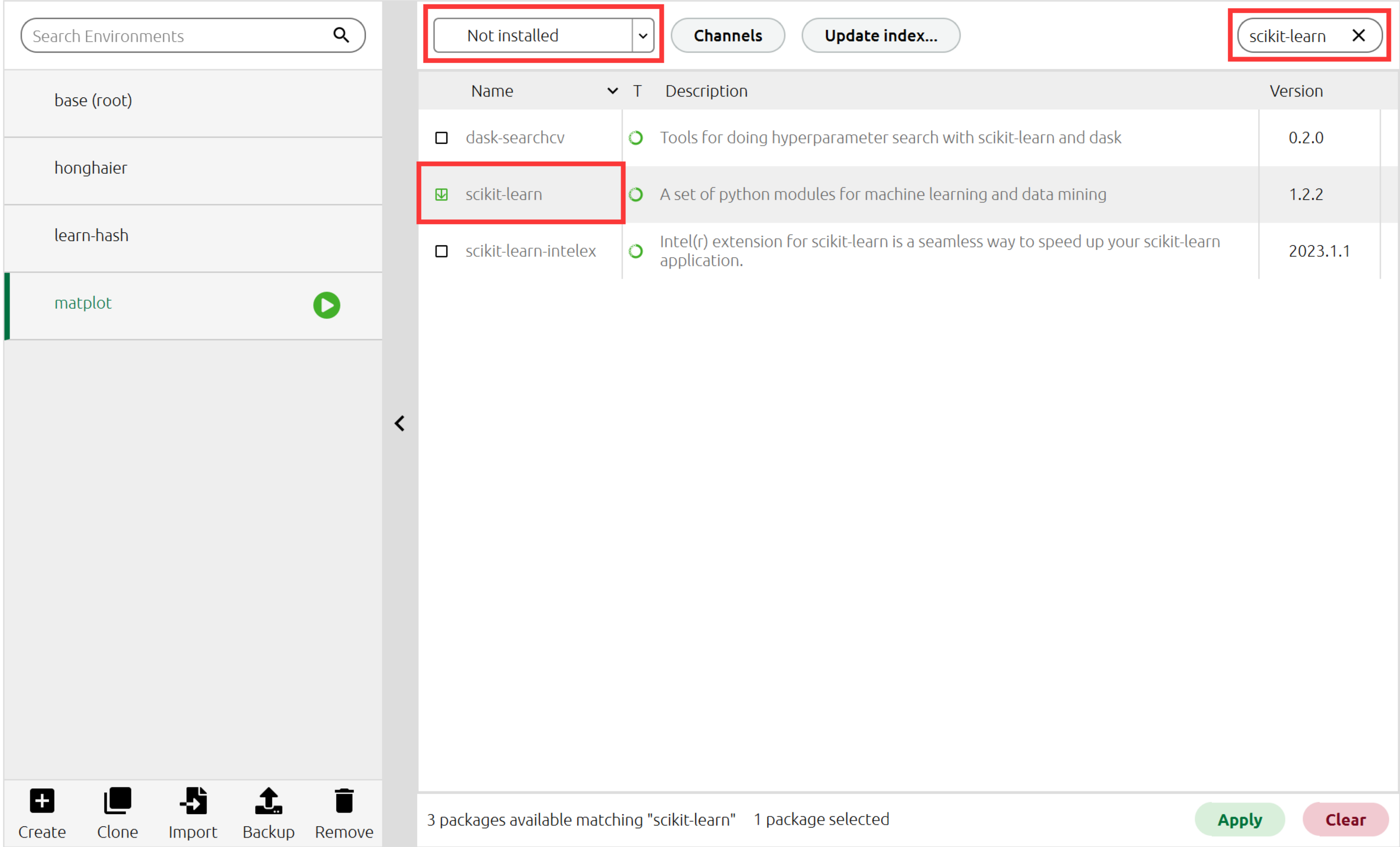


```
D:\Software\anaconda3\envs\matplot\python.exe
Overall accuracy:  0.8385214007782101


Process finished with exit code 0
```

实验步骤

- 安装机器学习库scikit-learn



Search Environments

- base (root)
- honghaier
- learn-hash
- matplot

Not installed

scikit-learn

	Name	T	Description	Version
□	dask-searchcv	○	Tools for doing hyperparameter search with scikit-learn and dask	0.2.0
☑	scikit-learn	○	A set of python modules for machine learning and data mining	1.2.2
□	scikit-learn-intelex	○	Intel(r) extension for scikit-learn is a seamless way to speed up your scikit-learn application.	2023.1.1

3 packages available matching "scikit-learn"   1 package selected

实验步骤

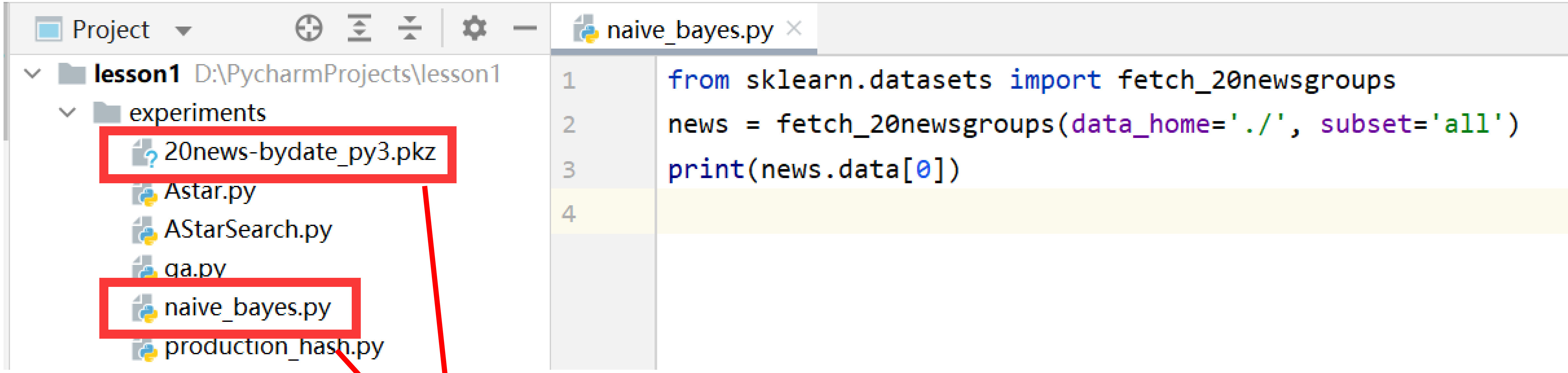
- 下载学习通里：资料/实验课ppt/20news-bydate_py3.pkz
- 新建一个python文件，输入以下代码：

```
from sklearn.datasets import fetch_20newsgroups
news = fetch_20newsgroups(data_home='./', subset='all')
print(news.data[0])
```

- 将20news-bydate_py3.pkz放在python文件的同目录下
- 运行上述python代码

实验步骤

- 例：

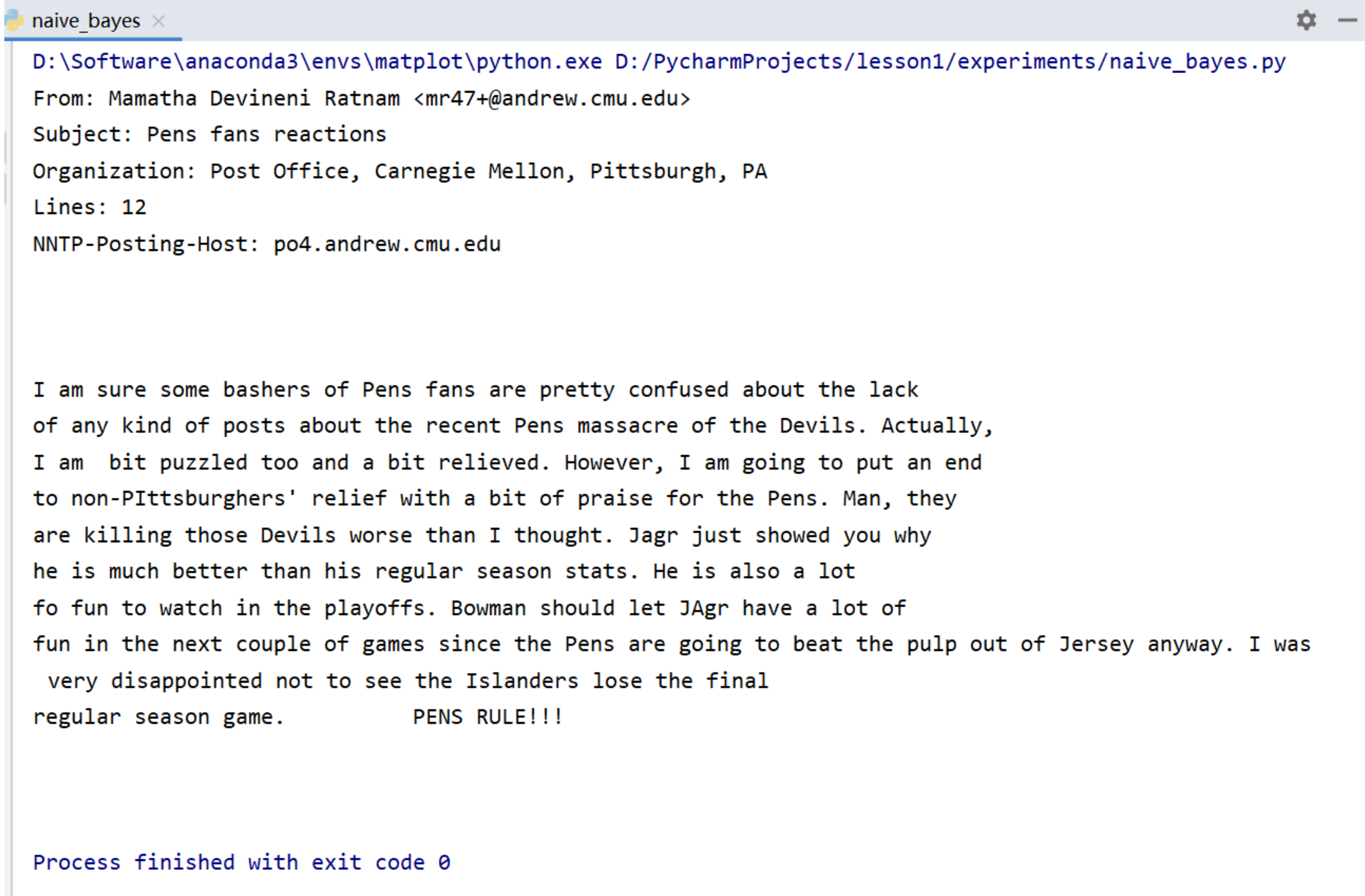


```python
from sklearn.datasets import fetch_20newsgroups
news = fetch_20newsgroups(data_home='./', subset='all')
print(news.data[0])
```

相同目录下！

实验步骤

- 成功示例：



```
D:\Software\anaconda3\envs\matplot\python.exe D:/PycharmProjects/lesson1/experiments/naive_bayes.py
From: Mamatha Devineni Ratnam <mr47+@andrew.cmu.edu>
Subject: Pens fans reactions
Organization: Post Office, Carnegie Mellon, Pittsburgh, PA
Lines: 12
NNTP-Posting-Host: po4.andrew.cmu.edu



I am sure some bashers of Pens fans are pretty confused about the lack
of any kind of posts about the recent Pens massacre of the Devils. Actually,
I am  bit puzzled too and a bit relieved. However, I am going to put an end
to non-PIttsburghers' relief with a bit of praise for the Pens. Man, they
are killing those Devils worse than I thought. Jagr just showed you why
he is much better than his regular season stats. He is also a lot
fo fun to watch in the playoffs. Bowman should let JAgr have a lot of
fun in the next couple of games since the Pens are going to beat the pulp out of Jersey anyway. I was
 very disappointed not to see the Islanders lose the final
regular season game.          PENS RULE!!!



Process finished with exit code 0
```

实验步骤

✓ 20newsgroups数据集简介

- ✓ 包含来自20个不同新闻组的文本数据。每个新闻组都包含多篇新闻文档，总共有18,846篇文档。
- ✓ 该数据集的文本数据涵盖了多个主题，包括科技、政治、体育、娱乐等。每个文档都被分配了一个特定的标签，表示其所属的新闻组类别。

朴素贝叶斯

$$P(X \mid A,B,C) = \frac{P(X)P(A \mid X)P(B \mid X,A)P(C \mid X,A,B)}{P(A,B,C)}$$

$$= \alpha \cdot P(X)P(A \mid X)P(B \mid X,A)P(C \mid X,A,B)$$

朴素：假设特征A，B，C之间两两独立

原条件概率公式可转化为：

$$\alpha \cdot P(X)P(A \mid X)P(B \mid X,A)P(C \mid X,A,B)$$

$$= \alpha \cdot P(X)P(A \mid X)P(B \mid X)P(C \mid X)$$

实验步骤

- 导入需要使用的方法:

```
# 朴素贝叶斯分类器
from sklearn.naive_bayes import MultinomialNB

# 文本向量化
from sklearn.feature_extraction.text import CountVectorizer

# 数据集分割
from sklearn.model_selection import train_test_split

# 计算分类准确度
from sklearn.metrics import accuracy_score
```

实验步骤

- 定义news_predict方法，用于训练朴素贝叶斯分类器，并对测试集中的样本进行分类：

```
def news_predict(train_sample, train_label, test_sample):
    vectorizer = CountVectorizer()
    X_train = vectorizer.fit_transform(train_sample)

    # 训练朴素贝叶斯分类器
    NB_classifier = MultinomialNB()
    NB_classifier.fit(X_train, train_label)

    X_test = vectorizer.transform(test_sample)

    # 预测测试样本的类别
    y_pred = NB_classifier.predict(X_test)
    return y_pred
```

实验步骤

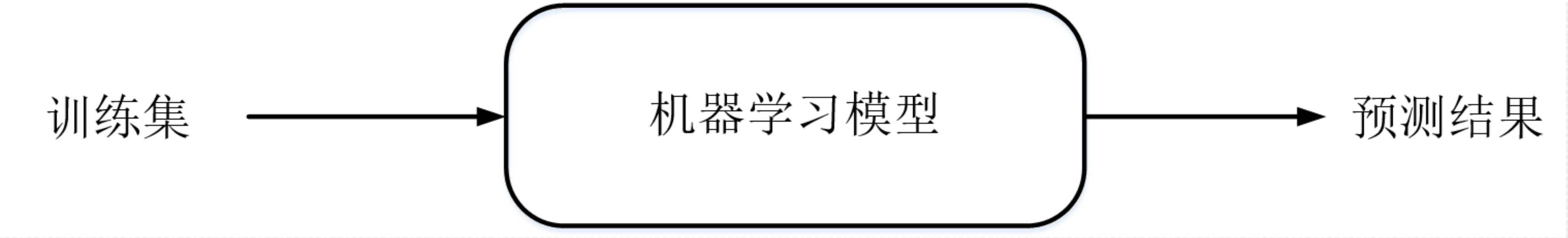
```
# 文本向量化
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_sample)
```

- 数据集 $Dataset\ =\ \{Training\ set,\ Test\ set\}$
- 一般训练集占80%，测试集占20%
- 样本 $\overrightarrow{x_i} = (a_i, b_i, c_i, \dots)$，其中$a_i, b_i, c_i$均为特征(feature)
- 由于原始数据不是向量，因此要先对每个样本进行向量化

实验步骤

```
# 训练朴素贝叶斯分类器
NB_classifier = MultinomialNB()
NB_classifier.fit(X_train, train_label)
```

训练阶段：

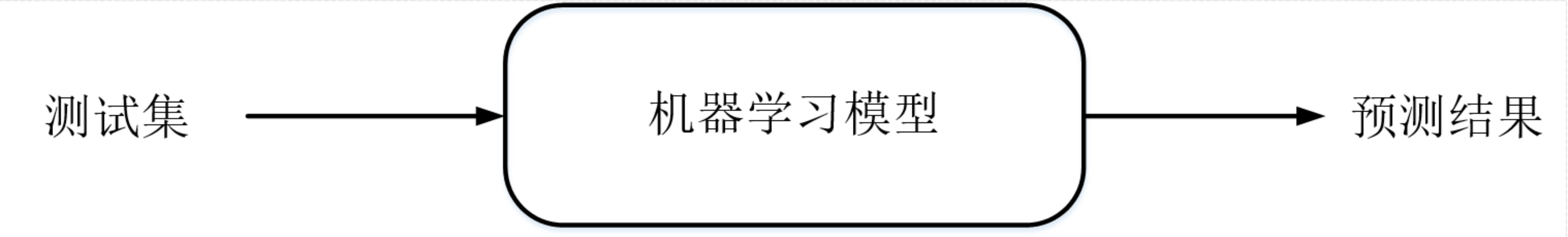


- 设计一个损失函数 $loss = loss_func($预测结果，真实结果$)$
- 通过优化算法降低loss，使得预测结果和真实结果尽可能接近
- 优化loss的过程 = 调整机器学习模型的参数

实验步骤

```
X_test = vectorizer.transform(test_sample)
# 预测测试样本的类别
y_pred = NB_classifier.predict(X_test)
```

测试阶段：



- 计算 $\text{acc} = Accuracy($预测结果，真实结果$)$，以评估模型性能

实验步骤

```
# x表示新闻文本，y表示类别(标签)
x = news.data
y = news.target

# 把数据集分成测试集和训练集
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

# 调用之前写好的news_predict方法
y_predict = news_predict(x_train, y_train, x_test)

# 计算总体分类准确度
acc = accuracy_score(y_test, y_predict)
print('Overall accuracy: ', acc)
```

结束语

谢 谢！