**深 圳 大 学 实 验 报 告**

**课程名称：­ 概率论与数理统计**

**实验项目名称: Bayes Classifier in Natural Language Processing**

**学院： 电子与信息工程学院**

**专业： 电子信息工程**

**指导教师： 陈昌盛**

**报告人： 杨烨 学号**： **2022280380**

**班级： 06**

**实验时间： 2023年11月10日**

**实验报告提交时间： 2023年11月30日**

**教务处制**

|  |
| --- |
| Aim of Experiment:   1. Familiar with the Bayes theorem. 2. Understand the implementation of the Bayes theorem in python. 3. Know how to use Naive Bayes for a practical task, e.g., text classification. 4. Master basic python programming, familiar with class inheritance, better handle dictionaries, lists and other data, and learn how to import training sets, import files; 5. Master the principle of naive Bayes and use naive Bayes to make spam classifiers; 6. Master how to handle zero probability data using Laplacian smoothing techniques and how to do it programmatically. |
| Experiment Content:  The main contents of naive Bayes spam classification experiment include data collection, data import, data preprocessing, feature extraction, model selection and training, model evaluation, parameter optimization, prediction and application. |
| Experiment Process：   1. Import data set:   Put the given data set in the file root directory; Import and store in the corresponding list;  spam\_files = os.listdir(os.path.join(DATA\_DIR, subfolder, 'spam'))     1. In the imported data set, we label spam 1 and non-spam 0;   Call function: X, y = get\_data(DATA\_DIR)创建类SpamDetector\_1：     1. Three functions are defined under this class:   The clean(self, s) function cleans text data;  The tokenize(self, text) function splits the text into a list of words;  The get\_word\_counts(self, words) function counts the frequency of words.  Then calculate the logarithmic prior probabilities of spam and non-spam：    Then walk through the training sample and corresponding labels and split the text:  If the term is not in vocab (vocabulary), it is added;  If the term is not in word\_counts[c] (the term counter for class c), initialize it to 0.0；     1. The selector is trained using a training set numbered after 100：      1. Define new classes to validate the test set：   Define related functions: calculate the probability of the word appearing in different scenarios; (Use Laplacian smoothing to calculate probabilities when necessary, i.e. zero probability events)  Where len(self.vocab) is the number of classes added to the smoothed denominator；    Add up the calculated P(content | spam) and P(content | normal mail)：    Add the prior probability:    Finally, if spam\_score > ham\_score, the value is 1, that is,spam     1. Test the training results：     Calculate the probability that a keyword appearing in a message is successfully tested as spam： |
| Data Logging and Processing:  Finally, I came to  the conditional probability of spam, the probability that the first and second rounds of mail are spam and non-spam, and the accuracy of the final judgment of the mail are: |
| Experimental Results and Analysis:  The experimental results are as follows:  In this experiment, my mail classification model based on naive Bayes theorem got 98% accuracy, and all the assertion codes were detected correctly.  Analysis:  The highly accurate mail classification model can provide users with a better mail filtering experience, help them quickly identify and process important emails, and filter out spam. This is very helpful for improving work efficiency and reducing the interference of spam to users. |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。