

Summer Class Python Course

Final Assignment

一、Python Code Duplication Checking Tool

1. Introduction

Design a simple tool to check duplications in Python code, which support batch checking of duplications and display of the results.

2. Basic Experiment Requirements

1. Use Tkinter, PyQt5 or other GUI libraries to design a GUI, or use a frontend and backend frame that Python supports to develop.
2. Develop a tool supporting batch code importing, check the duplicates of the code, show the duplication rate and so on.
3. The interface should be good-looking. Other functions can be added by yourself, but they should be simple and easy to use.

3. Required Task

1. Support batch code importing and one-to-multi inquiries.
2. Code duplication check function. Your program should give the duplication rate between pairs, and show the duplication rates in the GUI in a descending order in a reasonable and good-looking manner.
3. When clicking the entry of each result, display the contents of the two codes in detail, and highlight the similar and different parts of the codes.

4. Optional Task

1. Record each login time and display in GUI.
2. Add history record function: Store every import and check results, and show history check data without re-calculating the results.
3. Add plagiarism management function: For the duplication results of each pair of code, add functions of manual plagiarism tagging and exporting the list of suspected plagiarism behavior. It can be linked with Optional Task 2 and store the suspected plagiarism information in the history data.
4. Add function to export all plagiarism code.
5. Add function to choose 1-to-multi check or group self-check. When performing a group self-check, divide the results of plagiarism into groups and visualize.

二、Online question bank platform

1. Introduction

Design a question platform to help students complete the question self-assessment.

2. Basic Experiment Requirements

1. You need to use GUI libraries such as Tkinter and PyQt5 to design GUI, or use a front-end and back-end framework supported by Python for development and design.
2. The platform contains: a text extraction system that automatically recognizes PDF files or pictures, an editing system for text results, an automatic or manual question recording system, a question generating system, a question answering system, and a result analysis system.
3. The GUI is required to be presentable, and other functions can be added according to your capabilities, but they must be easy to use.

3. Required Tasks

1. Be able to identify relevant content by uploading PDF files or pictures of questions.
2. It has a simple yet beautiful UI to edit the identified results to adjust the inaccurate content.
3. Manually or automatically generate a list of questions based on the identified content, and then put them into the question bank.
4. The question generating system, which supports random questions and manual question selection.
5. The question answering system, which is convenient for users to test by themselves.
6. Result feedback function.

4. Optional Tasks

1. Using login and registration functions to distinguish users.
2. Wrong question mode. The user's wrong questions are recorded, which is convenient for users to check, and the questions can be set according to the wrong questions.
3. Favorite function. Record the user's favorite questions, and the questions can be set according to the favorite questions.
4. Reciting function. Present answers directly for users to recite.
5. The historical answer analysis function, which can analyze the user's previous results.

三、E-commerce sales forecast

1. Introduction

E-commerce sales forecasting is a major challenge facing the smart supply chain. Due to the influence of various factors, such as flow control, seasons, product categories, prices, promotions, etc., the sales forecast of online products faces huge challenges. How to accurately predict the future sales of online products and prepare for replenishment and allocation is a key problem to optimizing the present E-commerce productivity system.

This question first appeared in iFLYTEK open platform competition:

<https://challenge.xfyun.cn/topic/info?type=e-commerce-2022&option=ssgy>

2. Basic Experiment Requirements

Complete the algorithm design and sales forecast according to the requirements of the platform, and obtain the results after submitting on the platform. The ranking on the ranking board is used as the scoring standard of this task.











赛事概要

FAQ

排行榜

参赛团队

初赛

排名	参赛团队	分数	提交次数	最佳成绩提交时间
1	 7smbj1y2B	1833.28489	12	2022-07-21 23:24:05
2	 6o5SV8E32	1897.6729	21	2022-07-20 21:13:40
3	 OV3sKnTOE	1929.5732	14	2022-07-22 22:03:09
4	 戴卡奥特曼	2054.13752	7	2022-07-22 14:22:20
5	 Nvy3l6L2F	2095.93588	20	2022-07-18 11:14:04
6	 PG2BTIEGb	2144.46136	12	2022-07-20 09:38:49
7	 NlrmxU4P7R	2167.77279	5	2022-07-18 23:45:19
8	 K8DFQP0Ki	2189.00708	9	2022-07-19 20:26:11
9	 8rQqp1bUJH	2243.26906	11	2022-07-22 10:51:47
10	 mMlsBANIE	2514.75407	3	2022-07-16 16:50:26

3. Required Task

1. If you reach the top 70% of the total score, you can get 12 points. (The full score is 30 points)
2. If your ranking is between top 90% to top 70%, for every 2% reduction, 1 point will be deducted.
3. If your ranking is below the top 90%, you will get 0.

4. Selective Task

1. If your ranking is between top 70% to top 30%, 1 point will be added for each 4% ranking improvement.
2. If your ranking is between top 30% to top 15%, 1 point will be added for each 3% ranking improvement.
3. If your ranking is between top 15% to top 10%, 1 point will be added for each 2.5% ranking improvement.
4. If your ranking is above top 10%, you will get 30 points.

The ranking range described here is 70%~10% away, meaning $70\% \leq \text{grade} < 10\%$, i.e. $\text{front} \leq \text{rear} <$.

四、Task Scheduler

1. Introduction

As college life becomes richer by the day, homework and activities become more intertwined. Therefore, you need to design an application that help plans your own daily schedule, record tasks reasonably, and better plan your time.

2. Basic Experiment Requirements

1. You need to use GUI libraries such as Tkinter and PyQt5 to design a GUI.
2. Provide a working system with clear and concise functions such as a calendar, task status distinction, historical data review, task scheduling and others.
3. The interface has to be clear and concise, as well as presentable. Other functions can be added according to your capabilities, but the end user should be able to navigate the system with ease.

3. Required Task

1. Task adding system. The user should be able to create new tasks. Tasks should include a title, content description, deadline, importance, etc. At the same time, it can allow for modification and deletion.
2. Display daily task. Display the tasks that need to be completed every day.
3. Confirmation of task completion. When a task is completed, the user can delete the task through some method, such as checking a tick box.
4. Calendar system. User can view the task schedule for each date of a month by viewing the calendar.
5. Task status distinction. Distinguish between not started, in-progress, completed, and expired task statuses.
6. Task scheduling. Automatically schedule tasks by setting a task list and organizing free time.
7. User login system.

4. Optional Task

1. The to-do system can filter and display tasks according to time.
2. Task category division. Set task categories, such as sports, learning, etc.
3. Data recovery. The system can perform data analysis based on historical task completion data.
4. Support daily tasks. For daily tasks, you only need to set them once, and they will automatically appear in the task list every day within the specified time.
5. Weighted task scheduling. By making a task list and setting priority weights for tasks, the task plan can be automatically arranged while also allocating free time to rest.

五、Simple Data Visualization and Analytic Toolkit

Implement an elementary grade data visualization and analytic toolkit, which uses Python libraries to visual data based on the students' grade data set (data will be provided). If provided the ability, you can further enhance the toolkit to use machine learning or mathematics modeling related methods and analyze student achievement.

1. Basic Experiment Requirement

1. You need to use GUI libraries such as Tkinter and PyQt5 to design a GUI, or use front-end and back-end frameworks supported by Python for the development and design, as well as create a visualization of the data. To perform this task, you first have to perform data statistics to determine the total number of students, their nationality, places of birth, gender, etc. Then, use line charts, bar charts, pie charts, etc. to give a clearer picture of the statistics. Users should be able to understand the statistics of the provided data set with a clear and concise picture.
2. Data export function. The user should be allowed to select different types of data to export, such as male-female proportion pie-chart, etc.)
3. Academic performance evaluation visualization. Divide each student's grades into one of the three grades: **excellent**, **average**, **poor**. The grades may be related to factors such as the number of times a student raise their hands in class, the number of times absent, gender, whether the student is accompanied by their parents while studying etc. You can also present your data to teachers and teaching assistants the factors you think are closely related to grades, and explain the reasons (minimum requirements); if you can use mathematical related methods, such as fitting, regression and other methods of analysis, please do try to use mathematical methods in the programming and presentation (standard requirements); if you can use machine learning methods, such as decision trees and random forests to classify and model data to predict grades and find out if the factors are correlated, please use machine learning method in the programming and presentation (bonus marks).

2. Required Task

1. Complete the GUI design. The interface should be clear and concise. Users can choose to import the grade data set through the interface. Try to cover all the individual indicators of the data set according to the statistical data, such as country, gender, number of hands-up times in class, and number of absences.
2. Visually display the data. You can use bar charts, line charts, pie charts, etc. to display your data in the required part (1). Displaying data needs to be combined with the characteristics of the data. For example, it is more intuitive to use bar charts and pie charts for the proportion of males and females.
3. Show the overall grade performance according to class and academic year. It is required to be able to see the evaluation of each class's parents, students' grades, students' classroom activity and absences. Here, you can choose any method to display the statistical results, which can either be data display or visual display.

3. Optional Task

1. Distribute the student grades into four segments and plot a diagram. The four score segments are 85-100, 70-85, 60-70, and lower than 60.
2. The achievement display module can be displayed in more detail according to nationality, gender, etc.
3. Use mathematical or machine learning to create grade models. From that, analyze three factors that can impact a student's grade, and discuss the factors your understanding based on personal experiences.
4. Take 80% of the data set as training set and 20% as test set to verify your model. Then, remove the grade item, can your modeling predict student grades based on the preceding features? How much does the prediction differs from the actual situation?

For dataset please refer to the attachment **student_data_FIX.csv**

六、 Self-defined Question

1. Introduction

Define a question and propose a solution by yourself. For this, you are required to use GUI libraries such as Tkinter and PyQt5 to design a GUI, or use Python-related front-end and back-end technologies such as Django and Flask to complete the system design.

Before attempting the design, you need to write a proposal to state your intention and description of the problem. Please refer to the above five questions to get a rough understanding of the format. Send the proposal in PDF form to the email 17373255@buaa.edu.cn before 23:55 on the 26th of July. The research group will provide feedback before 23:55 the next day. After approval, the project can be commenced at your own accord.