

AI Challenge

SAMPLE QUESTIONS

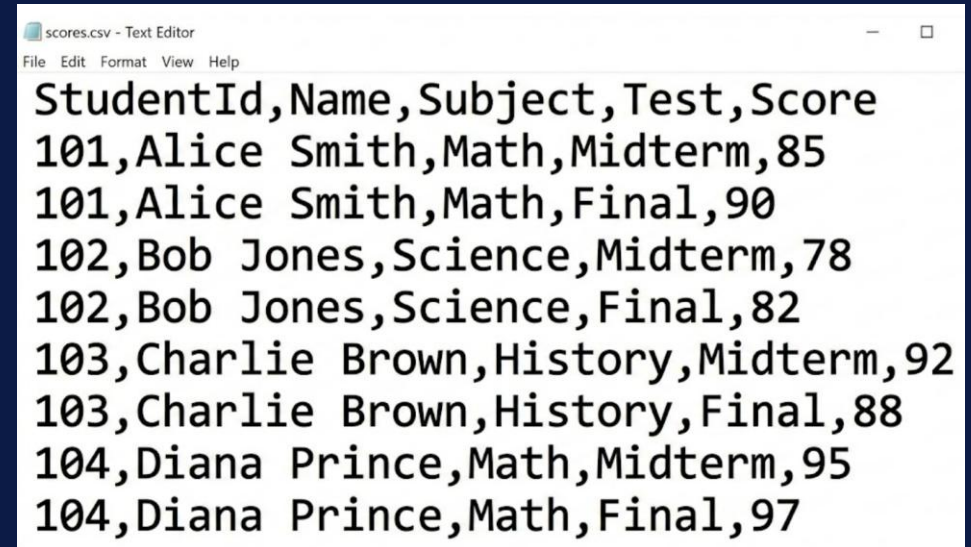
Tool: Visual Studio Code (Python)

Group 4: Adult (1st Secondary - 3rd Secondary)

Question 1:

You are building a school data analysis system to study students' test results. The scores are saved in a file called `scores.csv` as shown. To analyze the data using Python, you need to load this file into a Pandas DataFrame.

Which Pandas command correctly loads the data?



```
scores.csv - Text Editor
File Edit Format View Help
StudentId,Name,Subject,Test,Score
101,Alice Smith,Math,Midterm,85
101,Alice Smith,Math,Final,90
102,Bob Jones,Science,Midterm,78
102,Bob Jones,Science,Final,82
103,Charlie Brown,History,Midterm,92
103,Charlie Brown,History,Final,88
104,Diana Prince,Math,Midterm,95
104,Diana Prince,Math,Final,97
```

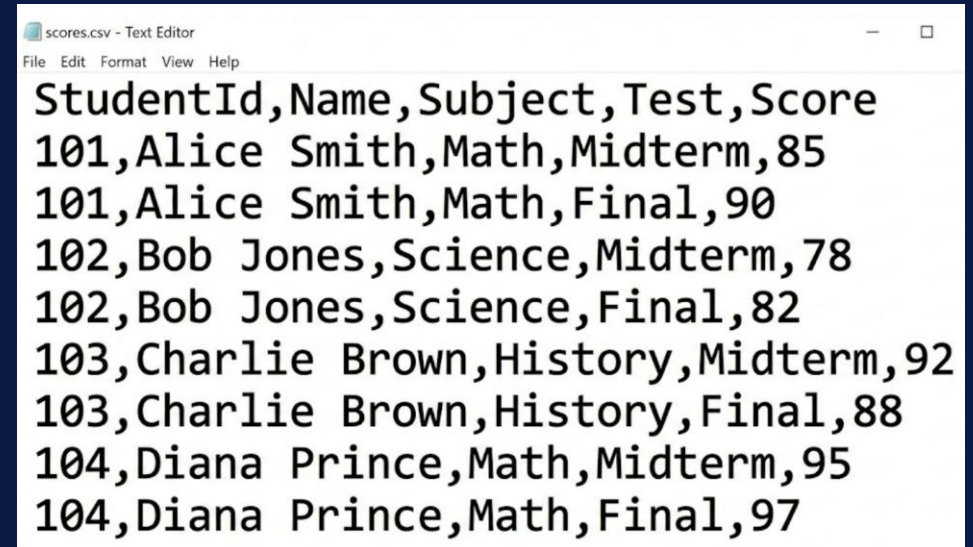
- (a) `data = pd.load_file('scores.csv')`
- (b) `data = pd.csv_read('scores.csv')`
- (c) `data = pd.read_csv('scores.csv')`
- (d) `data = pd.import_data('scores.csv')`

Question 2:

You are building a linear regression model to predict student grades in their finals. The dataset contains a column named “School_ID” which is just a random number and not useful for prediction. You want to remove it.

Which code snippet is correct?

- (a) `data.remove('School_ID')`
- (b) `data.delete('School_ID', axis=0)`
- (c) `data.drop(['School_ID'], axis=1)`
- (d) `data.clean(['School_ID'])`



```
scores.csv - Text Editor
File Edit Format View Help
StudentId,Name,Subject,Test,Score
101,Alice Smith,Math,Midterm,85
101,Alice Smith,Math,Final,90
102,Bob Jones,Science,Midterm,78
102,Bob Jones,Science,Final,82
103,Charlie Brown,History,Midterm,92
103,Charlie Brown,History,Final,88
104,Diana Prince,Math,Midterm,95
104,Diana Prince,Math,Final,97
```

Question 3:

You are developing a smart energy monitoring system for a building. The system should predict daily electricity consumption based on factors like temperature and number of people in the building. You have already loaded the data and created a linear regression model. Now, you need to train the model so it can learn the relationship between the inputs (features) and the electricity usage (target).

Which command trains the model using the data?

- (a) `model.train(features, target)`
- (b) `model.learn(features, target)`
- (c) `model.fit(features, target)`
- (d) `model.predict(features, target)`

Question 4:

You are using the Gemini API in streamlit. You want to convert the general AI assistant shown into acting like a professional French chef while responding to generate food recipes recommendations. Which line of code best sets this “persona”?



- (a) `prompt = "Hello AI, please give me a recipe."`
- (b) `prompt = "You are a professional French Chef. Create a recipe for..."`
- (c) `prompt = "I like French food. Cook something."`
- (d) `prompt = "Act like a robot and compute ingredients."`

Question 5:

You are starting a new ML project as shown in the figure and need to use Scikit-learn's model splitting tool.

Which import statement is correct?

```
# Importing libraries
import streamlit as st
import pandas as pd
# - sklearn splitting tool -
```

- (a) import split_data from sklearn
- (b) from sklearn.model_selection import train_test_split
- (c) import train_test_split from sklearn.linear_model
- (d) from pandas import train_test_split

Question 1:

You are building an AI car price predictor app. You want the user to select the car's transmission type from a specific list:

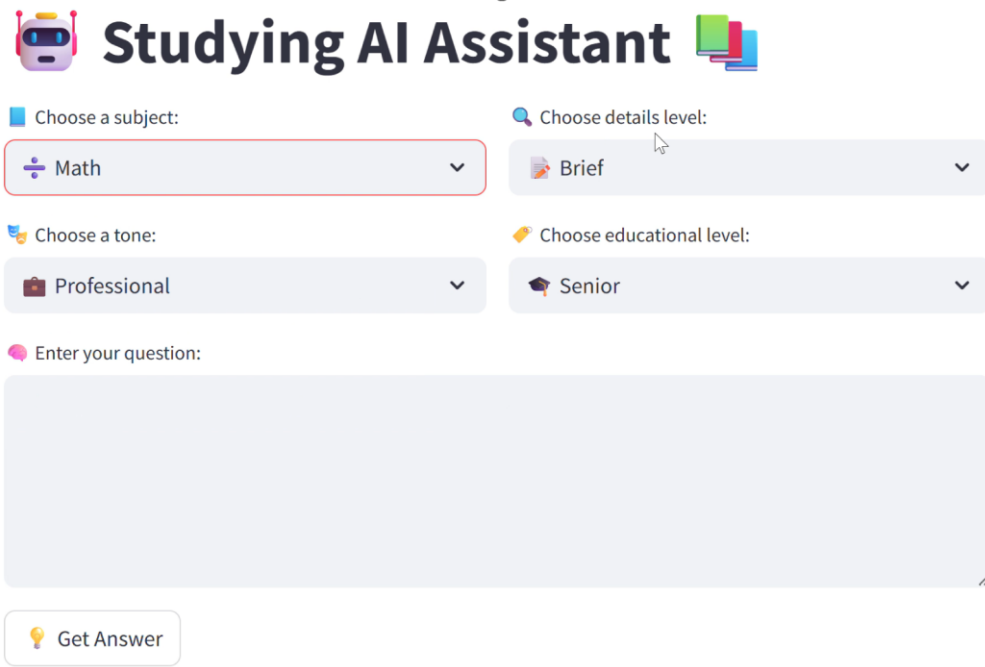
`["Automatic", "Manual"]`

Which streamlit widget is appropriate?

- (a) `st.text_input("Transmission")`
- (b) `st.slider(" Transmission ", ["Automatic", "Manual"])`
- (c) `st.selectbox(" Transmission ", ["Automatic", "Manual"])`
- (d) `st.button(["Automatic", "Manual"])`

Question 2:

You are building a studying AI assistant app as shown. The user chooses their educational level in a variable called `user_grade`. How do you correctly insert this into your prompt string?

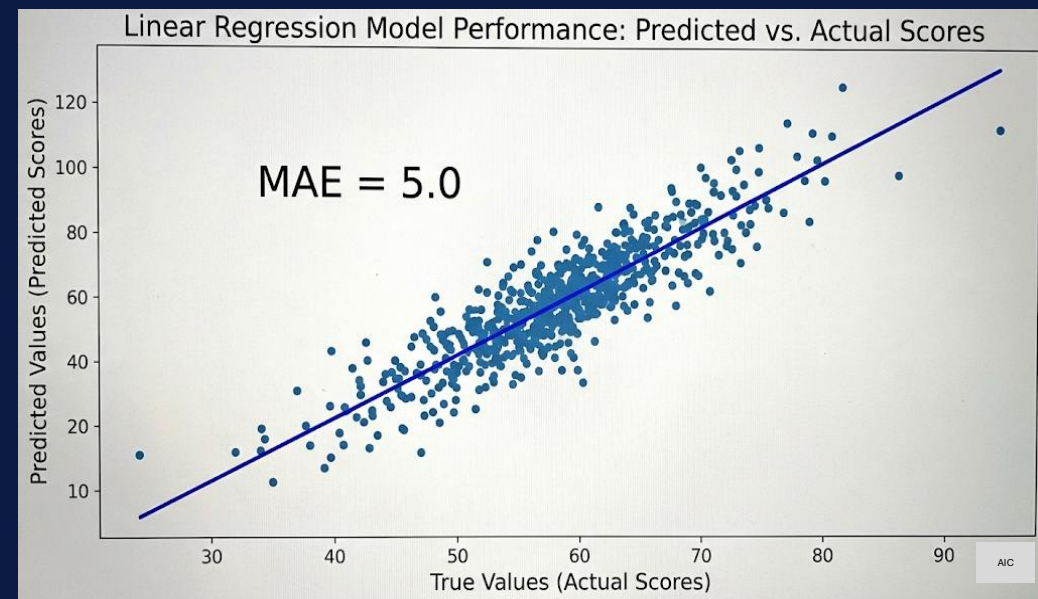


The screenshot shows the 'Studying AI Assistant' app interface. It features a header with a robot icon and the title 'Studying AI Assistant'. Below the header, there are four dropdown menus for user selection: 'Choose a subject:' (set to 'Math'), 'Choose details level:' (set to 'Brief'), 'Choose a tone:' (set to 'Professional'), and 'Choose educational level:' (set to 'Senior'). Below these menus is a large text input field labeled 'Enter your question:'. At the bottom of the interface is a button labeled 'Get Answer' with a lightbulb icon.

- (a) `prompt = "Explain this to a {user_grade} student."`
- (b) `prompt = f"Explain this to a {user_grade} student."`
- (c) `prompt = "Explain this to a" + user_grade + "student."`
- (d) `prompt = "Explain this to a [user_grade] student."`

Question 3:

You built a model to predict students test scores in their final exams using linear regression. You calculated the Mean Absolute Error (MAE) and got a value of 5.0. What does this likely mean?



- (a) The model is 5% accurate.
- (b) The model is 95% accurate.
- (c) On average, the model's prediction is off by 5 points.
- (d) The model failed to predict 5 students.

Question 4:

You are creating a Google maps trips time prediction tool that uses a linear regression model. You started splitting the data (Features, Target) as shown in the figure. But there's a major mistake in this code. However, there is a major mistake in this code, what's it?

```
f_train, f_test, t_train, t_test = train_test_split(  
    f, t, test_size=0.2)  
model.fit(f_test, t_test)
```

- (a) The test_size is too small.
- (b) The model is training on the test set (the small portion) instead of the train set.
- (c) The variable names are too short.
- (d) They forgot to print the result.

Question 5:

You start building an AI studying assistant app using Gemini API as shown in the figure. What's the benefit of using the `genai.configure()` function beside choosing the Gemini model?

```
# Choosing the model
my_key = 'AIzaSyAcVRhIxsNfbDspHScz230z3TsFpFhvix8'
genai.configure(api_key=my_key)
model = genai.GenerativeModel('gemini-2.5-flash')
```

- (a) It initializes the model settings as its context window
- (b) It retrieves all APIs you have and save them to be used by Gemini
- (c) It reset the API key for the model and hide it
- (d) It passes the API key for the model to work properly

Question 1:

You are making a study assistant app and wrote this code at its end. When you run the app and ask a question, instead of a clear answer, you see a confusing block of text on the screen that looks like this:
result: { candidates { content { parts { text: "Photosynthesis is..." } } } safety_ratings { ... } }. What's the reason?

```
if st.button('💡 Get Answer'):  
    prompt = f'''  
    You are an AI studying assistant helping a {edu_level}  
    level student with {subject}.  
    Answer in a {tone} tone and provide a {details}  
    explanation.  
    The question is:  
    {user_input}  
    '''  
  
    with st.spinner('Generating answer..'):  
        response = model.generate_content(prompt)  
        st.write(response)
```

- (a) The model returned a raw response object instead of extracting the text from it.
- (b) The `st.write()` function cannot display strings correctly.
- (c) The prompt is missing the question, so the model outputs debug info.
- (d) The app didn't import the streamlit library properly.

Question 2:

You are building a travel time prediction app. You wrote the following code to create input data for your linear regression model. When you run the app, you got this error:

“ValueError: Expected 3 columns, got 1”

What is the most likely the problem?

- (a) distance, speed, and passengers are Python lists instead of scalar values.
- (b) The pd.DataFrame is treating each input as a single column because the variables are None.
- (c) The model was trained on different column names, so the input DataFrame doesn't match the expected feature names.
- (d) The st.button() function doesn't return the values correctly, causing DataFrame creation to fail.

```
if st.button("Predict Arrival Time"):
    input_data = pd.DataFrame({
        'distance_in_km': [distance],
        'avg_speed_in_km': [speed],
        'no_of_passengers': [passengers]
    })

    predicted_time = model.predict(input_data)
    st.success(f"Estimated Time: {predicted_time:.2f} minutes")
```