Homework Assignment

The homework assignment is the primary technical evaluation mechanism. It has several parts. Expect to take a couple of days to complete this assignment; let us know if you need more time.

Programming Exercise

Build an application that allows users to interactively explore a dataset. This CSV file represents the data in question, however, you should design your application under the assumption that the actual data to be explored would be larger than can fit in RAM. At minimum, the application should support displaying the total number of viewers, by genre, from a set of cities defined by the user.

Take into account the following considerations in your design:

- The actual dataset being explored is larger than can fit in memory
- The application should provide some sort of reasonable, if elementary, user interface for the exploration
- Requirements change over time: will it be reasonably easy to extend or change your application to support additional interactions? For instance, can you filter the data by other parameters? Can you group the data by other fields? Can you change the metric being explored from "total number of viewers" to something else (e.g., average number of viewers)? It is not necessary to actually handle all possible cases, but it should be clear how the design would support change and extension.
    - **Answer:**
      Currently the application supports the minimum requirements specified. If we require to have more filtering options, the dataview flask Blueprint can be extended to support more filters. That would require adding more fields to the HTML form on the main page and updating the Flask Blueprint utilized by the main page to parse the filters and construct the required queries. In relation to the metric explored that could also be made using a HTML form input where the user could select one of multiple options like (sum or average).

- What if the underlying data model changed over time? What if the data model were made more complex and had additional tables, such as metadata about the cities or metadata about the programs (a simple such change might be supporting multiple genres per program)? Again, your application does not need to anticipate all possible changes in the data, but it should be clear how the design could support such changes.
    - **Answer:**
      In this case the database tables could be extended to support more metadata for the programs. We can create new tables with additional metadata and reference the programs we already have stored in the database. Then we can update the

application queries and views to query the additional metadata and display the new data.

- Suppose that your application is so wildly successful that other teams would like to integrate the same analytics your application produces into their own applications; consider how that sort of integration might be accomplished in your design.
    - **Answer:**
      In this case we would probably implement a REST API where all the data could be returned in some type of format that is easy to read programmatically. One example could be to return all data in JSON format so other teams could utilize this data.

The choice of programming language and any frameworks, libraries, or other tools (e.g., databases, etc.) is up to you, but please make every effort to make it clear how to install any dependencies and how to build your application so we can run it. (If this is complex, consider using Docker.)
Answer:
I decided to make the app using python Flask framework and sqlite3 as the database to store the data. All the instructions to run the application locally can be found in the README.md file on the root folder of the project repository.

Also, please anticipate how other developers would extend, modify, and use your code in the future.
Answer:
All development was made using git, in a real work environment this would allow the team to provide feedback as we introduce new features. I also followed the best practices explained by the python Flask framework documentation.

Perform a Code Review

Code review is an important part of our development process. Every piece of code we write undergoes code review by other members of the team before it is merged. Here is a small piece of code written in Python 3.8; what comments would you provide during code review? Write your comments the way you would if this was an actual pull request in GitHub/GitLab/etc.

```
import random


class TicTacToe:

    def __init__(self):
        self.board = [None]*9

    def make_move(self, row: int, col: int, player):
        position = col*3 + row
        self.board[position] = player
        return
```

```
def make_random_move(self, player):
    available_squares = []
    for i in range(9):
        if self.board[i] is None:
            available_squares.append(i)
    position = random.choice(available_squares)
    self.board[position] = player
    return

def _get_position(self, row, col):
    return col*3 + row

def _get_board_state(self, row, col):
    position = self._get_position(row, col)
    return self.board[position]

def _check_diagonal_win(self):
    center_square = self._get_board_state(1, 1)
    if not center_square:
        return None

    if ((center_square == self._get_board_state(0, 0)) and
            (center_square == self._get_board_state(2, 2))):
        return center_square
    elif ((center_square == self._get_board_state(0, 2)) and
          (center_square == self._get_board_state(2, 0))):
        return center_square

    return None

def _check_row_win(self):
    for row in (0, 1, 2):
        if (self._get_board_state(row, 0) ==
            self._get_board_state(row, 1) ==
                self._get_board_state(row, 2)):
            return self._get_board_state(row, 0)

    return None

def _check_col_win(self):
    for col in (0, 1, 2):
        if (self._get_board_state(0, col) ==
            self._get_board_state(1, col) ==
                self._get_board_state(2, col)):
            return self._get_board_state(0, col)

    return None

def check_win(self):
    for check_winner in (self._check_diagonal_win,
                         self._check_row_win,
                         self._check_col_win):
        winner = check_winner()
        if winner:
            return winner

    return None

def display_board(self):
    board = [square or ' ' for square in self.board]

    template = f"""
```

```
          {board[0]} | {board[1]} | {board[2]}
         --- --- ---
          {board[3]} | {board[4]} | {board[5]}
         --- --- ---
          {board[6]} | {board[7]} | {board[8]}
        """

        print(template)

        return


def main():
    ttt = TicTacToe()

    while True:
        ttt.display_board()
        print("Player X, choose your next move!")
        row, col = [int(square) for square in input().split(',')]
        ttt.make_move(row, col, 'X')
        winner = ttt.check_win()
        if winner:
            print(f"Player {winner} has won!")
            break

        ttt.make_random_move('O')
        winner = ttt.check_win()
        if winner:
            print(f"Player {winner} has won!")
            break

    return


if __name__ == '__main__':
    main()
```

Recommend a Technology

Imagine you intend to run the application you wrote for your programming exercise in production as a web-delivered service, and it will be accessible to the general public, so may need to support a fair amount of user load. (You don't need to assume that the users will have accounts or that you need to save any user state.)

What technologies or platforms would you recommend and why? What changes might you have to make? What additional considerations would you need to think about to operate and maintain the system?

Answer:

- Production deployment:
    - For this application I would recommend running it on a Kubernetes cluster if available and use a LoadBalancer with SSL certificate as the endpoint for customers. The Kubernetes pod should be configured accordingly based on the load expected and if possible configure some policies to scale up/down based on

requests being received. Running applications in Kubernetes is less expensive than reserving a whole Virtual Machine to deploy your application (The K8s cluster management is something that should be considered). AWS EKS service could be an option to deploy the team applications.

    - If Kubernetes isn't an option, it could be configured on a Platform like AWS with ec2 instances on an autoscaling group with a launch configuration that contains the instructions to configure the application. The autoscaling groups instances should be registered with a target group that can be attached to a load balancer.
        - Another option would be to run the application using docker instead of bare ec2 instance, or use a service like Amazon Fargate.
    - From the above, Kubernetes would be the most cost effective option with better resource utilization.

- Changes, improvements and considerations:
    - I would need to change the database to something that aligns with the project requirements. If using relational databases I would use a service like Amazon RDS for the database. Other options would be to use something like Elasticsearch and AWS managed ElasticSearch Service, MongoDB or data storage options more suitable for data analysis.
    - I would add more data visualization and more functionality to filter the data.
    - If other teams would be accessing the data the application would probably require an API that would allow other teams to access the data easily programmatically.
    - I should consider if the application would need to be publicly accessible or internally within the company and based on that decide on what cloud provider or datacenter I would like to deploy my application. The best location is always close to your customers for less latency. In relation to services open to the internet I would need to make sure that the application resources that would be open to the public are secure following best practices.
    - I would need monitoring and alerting for a production deployment. Either utilizing AWS Cloudwatch or 3rd party tools that would help with continuous and automated monitoring.
    - Another thing that should be considered is secrets management and how to handle the databases passwords or secrets like API Keys. Leveraging tools like AWS KMS or Hashicorp Vault should help with this.
    - The other important addition would be a continuous integration / continuous delivery pipeline that would allow the development and deployment of the application to be automated with all the required tests to pass before deployment (Gitlab runner or Jenkins pipelines should help on achieving this)
    - Lastly I would also use Infrastructure as Code with version control. Either using Helm Charts, Terraform or Ansible Playbooks or a combination of all for the application configuration.

Recommend a Process

Imagine you are working on a Scrum team doing two week sprints. The team finds that, quite often, planned stories aren't being completed within the sprint and are "rolling over" into the next sprint.

- Is this really a problem? If so, why, and what consequences might it have?

Answer:

Yes it is a problem, when we take work on a sprint the goal is to complete that work on that sprint so we can move forward with the work planned for the next sprints. If we keep "rolling over" other work would be affected. This could lead to missing a deadline established with a customer.

- What processes or strategies might the team adopt to improve?

Answer:

We need to evaluate the reasons why we keep rolling work over the next sprint, maybe we need to further divide the work so we can have smaller tasks that would help us focus more. Sometimes the work requirements are poorly specified, we might need to work on the specifications of certain tasks so when we decide to take them on a sprint we don't waste time looking at how the task should be done. One last thing to check is the workload we take on a sprint, maybe we are taking too much work when we could be focusing on less work with higher priority.