Team Project

```
$ echo "Data Sciences Institute"
```

Today

1. Team Project Overview

2. Developing a Project Plan

3. Project Goals and Requirements

4. Working as a Team

Team Project Overview

Goal: To enhance your portfolio by showcasing this team project as evidence of your ability to deliver real-world value to an employer.

Why This Module Matters

- This module is designed to help you create a **high-impact project** that showcases your skills in data science and machine learning.

- A **strong team project** can enhance your professional portfolio, making you more competitive in job applications.

- Your project will:
  - Demonstrate **your technical abilities and problem-solving skills**.
  - Serve as a **portfolio piece** for recruiters, hiring managers, and data science professionals.
  - Showcase **what you've learned throughout the certificate**.

- Once completed and refined based on feedback, you can **add it to your resume, GitHub, and LinkedIn** to increase your professional visibility.

What is a Good Portfolio?

A good portfolio showcases your best work and highlights skills that are highly valued by employers in data science and machine learning. It demonstrates your ability to handle real-world tasks, making you an attractive candidate for employment.

Personalize Your Portfolio:

- **Remove unnecessary content:** As you work on your project, ensure that any unused files or folders are deleted to keep your repository clean and professional.

- **Highlight your unique contributions and skills:** This personalization shows employers that you're not just completing assignments but are engaged and innovating on your projects.

Demonstrating Technical and Business Skills

- **Demonstrate Coding Proficiency**: All projects should showcase your ability to write clean, efficient, and well-documented code.

- **Showcase Communication Skills**: All projects should effectively communicate complex data-driven insights in a non-technical way.

Your ability to document code clearly and communicate effectively can drastically reduce onboarding times for new team members and help non-technical stakeholders make informed decisions, thereby enhancing overall project success and team efficiency.

Types of Projects to Include in Your Portfolio

Including a diverse range of projects in your portfolio can significantly enhance your appeal to potential employers. Consider including a variety of project types to demonstrate the breadth and depth of your data science skills:

1. **Data Cleaning Project:** Show your ability to prepare data for analysis.

2. **Data Storytelling and Visualization Project:** Highlight your skills in interpreting and presenting data in compelling ways.

3. **ML Modeling:** Demonstrate your proficiency in building and tuning models.

4. **Group Project:** Showcase your teamwork and collaboration skills. (already doing this! 🥳)

For more, read "How to Create a Project Portfolio for Data Science Job Applications"

Selecting Projects That Showcase Your Skills

Choosing the right projects for your portfolio is crucial. Each project should:

- **Solve Real Problems**: Use actual datasets to address genuine issues in your field.

- **Demonstrate Industry Relevance**: Select projects that are pertinent to your specific area, such as marketing analytics for marketers or predictive maintenance for engineers.

- **Provide Actionable Insights**: Focus on projects that deliver clear, practical outcomes that demonstrate your ability to impact real-world scenarios.

These criteria ensure that your projects not only highlight your technical skills but also your understanding of and adaptability to industry-specific challenges, making you a valuable candidate to potential employers.

Showcase Projects That Use Diverse Data Types

Enhance your projects by effectively using a mix of structured, unstructured, and time series data:

- **Structured Data**: Employ databases for clear, quantifiable insights.

- **Unstructured Data**: Add depth with text, images, or videos.

- **Time Series Data**: Utilize data in sequential order for trend analysis and forecasting.

Select data types strategically to align with your project's objectives. Ensure each type contributes to a clear and coherent narrative, avoiding unnecessary complexity to maintain focus and utility.

Developing a Project Plan

✅ Developing a Project Plan

1. **Understand the business context**: "We want to increase customer engagement by 10%."

2. **Identify an opportunity**: "Customer segmentation will enable targeted marketing campaigns, which should increase customer engagement."

3. **Scope your analysis**: "We will use feature engineering to create segments based on purchase history and demographic data."

4. **Develop your solution**: "We will k-means clustering with Python, and determine the optimal number of clusters using the elbow method."

5. **Present results and recommendations**: "We predict that applying targeted marketing based on 5 different consumer profiles will increase customer engagement by 10%."

✅ Developing a Project Plan

1. Understand the business context.

2. Identify an opportunity.

3. Scope your analysis.

4. Develop your solution.

5. Present results and recommendations.

📋 To keep in mind:

- Are your insights *actionable?*

- How robust is your analysis? *Quantify!*

- What are the *risks* and *unknowns*?

Project Goals

Project Goals

- **Good Code & Structure** → Code should be well-commented, clean, and follow a logical structure. It should be easy to read and maintain.

- **Strong Documentation & Presentation** → The README should be clear, well-written, and explain the dataset, findings, and methodology. The project should be easy to understand for both technical and non-technical reviewers.

- **Application of Module Teachings** → Projects should showcase key technical skills, such as regression modeling, deep learning models, data visualizations, or strong analysis of sampling techniques.

- **Effective Team Collaboration** → Teams should follow best practices for Git (small commits, branches, pull requests) and actively participate in stand-ups and progress updates.

Project Requirements

Final Project Submission

Your final project will be evaluated on the following criteria:

1. Each team member must have created a pull request, and reviewed and merged a different pull request.

2. (a) For **Data Science** teams, your project must include at least one visualization that presents new insights into the chosen dataset.

   (b) For **Machine Learning** teams, your project must include at least one machine learning model that you have developed and implemented to obtain new insights.

Final Project Submission

3. Each project's README should describe the project objective, the final outcome of the project, the key business takeaways, and describe your team's approach to working collaboratively. It should also demonstrate thoughtful consideration of the guiding questions.

Final Project Submission

4. Your project must include ***one completely reproducible feature***. This could be a single visualization, training and storing a model, or using a previously trained model to predict results, and the value of the feature should be described in your README. It could be formatted as a self-contained Jupyter notebook or a script, however anyone new to your project should be able to reproduce it based on the instructions provided, including any software, data, or file dependencies.

Final Project Submission

5. Each team member must record a 3-5 minute video reflecting on your experience. You may each choose where to host your own video, however *it should be public and a link to each team member's video should be included in your project README*. This video is meant to be an asset to your portfolio, and should be available for prospective employers. Your videos should answer the following questions:
   - What did you learn?
   - What challenges did you face?
   - How did you overcome those challenges?
   - If you had more time, what would you add?
   - What strengths do you bring to a team environment?

Expectations for Week 1

After Week 1, you will be evaluated on your project's README file. By this point, it must include a detailed project proposal. This should include the business motivation for your project, the dataset you have chosen to use, and any risks or unknowns you have identified.

Project Examples

- Pharma Spending by Countries (Cohort 4)

- Breast Cancer Wisconsin (Cohort 4)

- NYSE Prediction (Cohort 4)

- Bank Marketing Campaign (Cohort 5)

- Customer Purchasing Behaviours (Cohort 5)

- Stroke Prediction (Cohort 5)

Working as a Team

GitHub

Effective GitHub Repository Organization

- Ensure your GitHub repository is neatly organized; avoid unused or empty folders and ensure each folder has a clear purpose.

- Use READMEs in key folders (like `src`, `iac`, `backend`, `frontend`, etc.) to detail their contents and purpose, aiding clarity for complex sections.

- Avoid READMEs in simple folders (like `images`, `docs`, etc.), unless there's specific information that needs to be explained.

Managing your Repository as a Team

- Do not commit directly to the **main** branch!

- Each feature should be developed in its own branch, then merged into **main** when it has been completed and tested.

Merge Conflicts

It is likely that you will encounter merge conflicts when combining your work as a team. Make sure you've reviewed the Git material regarding how to resolve merge conflicts properly!

Working as a Team

Documentation

Crafting a Comprehensive Main README File

- **Purpose & Overview:** Introduce the project with essential details, concise description and a project objective.

- **Goals & Objectives:** Articulate what the project aims to achieve.

- **Techniques & Technologies:** Highlight the tools and methods used.

- **Key Findings & Instructions:** Summarize outcomes and provide setup instructions.

- **Visuals & Credits:** Enhance with visuals; acknowledge contributors.

Writing Clear Documentation (README & Comments)

While **you can have multiple README files** in your repository, include only one README file per folder to keep things clean and simple. This structure showcases your ability to manage and present complex information effectively, increasing your appeal to potential employers.

Effective commenting enhances code readability and maintainability, crucial for collaborative environments.

If someone with no tech background reads your code, they should get the essence of what you are doing and how the code flows. This practice not only aids in understanding but also facilitates smoother transitions and updates within team projects.

📰 Best practices for writing code comments

Example

❌

```
function calculateTotal(price, quantity) {
    return price * quantity;
}

let totalPrice = calculateTotal(25, 5);
console.log(totalPrice); // Output: 125
```

✅

```
// Calculates the total cost by multiplying the price per item with the quantity
function calculateTotal(price, quantity) {
    return price * quantity;
}

// Example usage: Calculates the total price for 5 items at $25 each by multiplying the price per item ($25)
// with the quantity (5), and stores the result in the totalPrice variable.
let totalPrice = calculateTotal(25, 5);
console.log(totalPrice); // Output: 125
```

Project Structure

- **One** person from each team must host the primary Git repository. Other team members should clone that repository to work on it.

- There is no hard requirement for the folder structure of your project, but best practices should be followed.

Working as a Team

The following two weeks will fly by quickly! How will you ensure that you can accomplish everything you need to as a team in this timeframe?

Working as a Team

1. **Clear objectives and milestones**. Milestones should be set working backward from the final project goal and deadline.

2. **Accountability**. Every action item should have someone assigned. (Of course, the assigned person can change if workloads become unbalanced.)

3. **Communication of roadblocks**. It is just as important to communicate roadblocks and failures as successes. The faster problems are identified, the faster they can be addressed.

Working as a Team

4. **Tracking**. Tracking your tasks is necessary to objectively assess your progress as a team. The data-driven mindset applies here too! If you fall behind, update your plan accordingly.

5. **Documentation**. Team members should be able to understand and continue each others' work where necessary.

Prepping You for the Industry

- Throughout the certificate, we've introduced the analogy:
  - **Participants → Junior Data Professionals**
  - **Learning Supports → Senior Data Professionals**
  - **Technical Facilitators → Product Managers**
- We're now taking this further by incorporating **stand-up meetings**, a common practice in professional data teams.
- Stand-ups help teams **stay aligned, communicate blockers, and improve efficiency**.
- By practicing stand-ups, you'll develop **strong communication and collaboration skills** that are essential in the workplace.

Daily "Standups"

- Each day, a member of the DSI instructional team will guide your team through a stand-up.

- Stand-ups are **quick, structured check-ins** that help teams stay on track and remove obstacles.

- The goal is not to compete for who did the most work—it's to ensure the entire team is working effectively and efficiently.

- This is a great opportunity to help your teammates and **resolve blockers early**.

- Stand-ups should take no more than **10 minutes**.

Daily "Standups"

- Each team member should describe:
  - What did you work on yesterday?
  - What will you be working on today?
  - Are you unsure about any of your tasks?
  - Are you blocked by anything?

Module Schedule

Each day will include "standup" meetings with your team, and either the Technical Facilitator or Learning Support.

Day 1 (Tues): [no session]

Day 2 (Wed): 1 hour of content delivery, 1.5 hours of co-work.

Day 3 (Thurs): 0.5 hours of content delivery, 2 hours of co-work.

Day 4 (Fri): Co-work.

Day 5 (Sat): Co-work.

Module Schedule

Day 6 (Tues): Review + co-work + Standup Meetings.

Day 7 (Wed): Co-work + Standup Meetings.

Day 8 (Thurs): Case Study.

Day 9 (Fri): Co-work.

Day 10 (Sat): Project showcase.

Goals for Today

- Start developing a business case.

- Choose your dataset.

Questions?