Team Project

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

Today

1. Team Project Overview

2. Developing a Project Plan

3. Project 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.

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.

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. (*examples in the next slide*)

- **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.

6

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 database data 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.

Project Examples

TTC Transit Delay Project

Mortgage Risk Assessment Project
(Cohort 3)

Mexican Government Report Text Analysis
(independent)

Project Requirements

- **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.

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.

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!

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
```

14

Better Documentation - STAR Method

- The STAR method is typically used for answering interview questions, and applying this structure to your project overviews can be highly effective.

- By organizing your projects using the Situation, Task, Action, Result format, your portfolio essentially speaks for you, conducting a virtual interview with potential employers even in your absence.

- This approach ensures that employers can clearly understand the value and impact of your work, even without direct interaction.

- Remember, the results don't have to be ground breaking, anything that showcases your abilities in data science/machine learning, even "**learning**" counts as an important result.

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.

Module Schedule

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

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

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

Day 3 (Thurs): Co-work.

Day 4 (Fri): Co-work.

Day 5 (Sat): Co-work.

Module Schedule

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

Day 7 (Wed): Co-work.

Day 8 (Thurs): Co-work.

Day 9 (Fri): Co-work.

Day 10 (Sat): Project showcase.

Goals for Today

- Start developing a business case.

- Choose your dataset.

Questions?