Closely pay attention to the feedback that you get on each of your deliverables. Doing so will help you both improve your work and alert you to any necessary adjustments.

Here are the tasks for all the segments:

* **Segment 1: Sketch It Out:** Decide on your overall project, select your question, and build a simple model. Then prototype your idea by using either CSV or JSON files to connect the model to a fabricated database.
* **Segment 2: Build and Assemble:** Train your model, build the database that you'll use for your final presentation, and create both your dashboard and your presentation.
* **Segment 3: Put It All Together:** Put the final touches on your model, database, and dashboard. Then create and deliver your final presentation to the class.

You’ll integrate many aspects of the course, which you've already encountered, into this final project. For example, you’ll use Python to clean, prepare, and explore the data and to complete the initial analysis. And, you can use Python libraries. Tableau, and JavaScript libraries—like Data-Driven Documents (D3) and Plotly—to create visuals that help tell your data story.

To store your cleaned data, you’ll use database integration (Postgres, MongoDB, or SQLite). You’ll also implement machine learning to enhance your topic. Your work will need a showcase—so use tools like Tableau or JavaScript to build a dashboard that presents your results. Finally, you should prepare and deliver a presentation that takes your class through your project, step by step.

To close, you'll write a short personal reflection after submitting your final project. Doing this self-assessment will provide you with the time and space to consider the experience, discuss the team dynamics, and think about how you’d describe the project to an interviewer.

Each segment and the closing self-assessment will be graded. The following table describes the grading breakdown:

Before diving into the first segment of your project, review the grading criteria for the project as a whole. Doing so will help you understand the end goal. You won’t deliver the requirements at once, however. Instead, we’ve divided the time that’s allocated for project work into three segments, and each has a specific deliverable that builds to the final product.

In total this project will account for 29% of your final grade in the course. Each segment of the project will be worth 100 points for a total of 300 available points. Each segment will also have separate requirements for building the content of the project and the presentation for the project. The summaries of the overall requirements for content and presentation are laid out below.

**Content Requirements (170 points)**

The requirements for earning the full 170 points across the three deliverables are relatively open ended. Creating a machine learning model, a database, and a dashboard is certainly no small feat! But, the topic, the technologies that you use, and the project goals are completely up to you. Your grade won’t be based on how well your model performs or whether your project is completely ready for thousands of people to use. Instead, you’ll be graded on whether you and your team demonstrate an effective use of a select few technical building blocks. This should allow your team to look beyond the requirements and dream big. But make sure that in each segment, your work always starts with what’s required before moving onto other goals.

To meet the requirements, the project and repository must contain the following:

* A detailed README.md file that includes the project status, images, descriptions, and results.
* At least 12 total commits per team member.
* A machine learning model, including a confusion matrix and an accuracy score.
* A database that stores the data for the project and that contains at least two tables or collections
* A dashboard that has at least one interactive element.

**Presentation Requirements (130 points)**

The presentation is worth 130 points across the three segments. And, 70 of those points will come from the third deliverable of the project, where you’ll showcase your work. This will be your chance to share a polished presentation that’s ready for either an interview or the boardroom.

Remember that you won't be graded on whether you've built the best model. Instead, you'll be graded on how you communicate your process: How did you choose the model, and why? What roadblocks did you encounter (roadblocks always occur!), and how did you solve them? And critically, can you illustrate all these points in a way that’s engaging for your audience?

To meet the requirements, the presentation must tell a cohesive story about the project and include the following:

* The selected topic and the reasoning for that selection.
* A description of the data source.
* The questions that the team planned to answer with the data.
* A description of the data exploration phase of the project.
* A description of the analysis phase of the project.
* The technologies, languages, tools, and algorithms that the team used throughout the project.
* The results of the analysis.
* Any recommendations for a future analysis.
* Anything that the team would have done differently if they had more time.

Additionally, the following requirements must be met:

* All the team members present in equal proportions.
* The team demonstrates the interactivity of the dashboard in real time.
* The presentation falls within any time limits that the instructor provides.

**Written Assessment (100 points)**

After the project is complete, each team member will submit a written assessment. This assessment is graded separately from the final project and will account for 1% of your final grade. There are 10 available points for the assessment.

**Self-Assessment (40 points)**

Your self-assessment should be a few paragraphs in length and meet the following requirements:

* **Cohesive written analysis:** Describes the role or roles that you played over the course of the project and the contribution of each of those roles to the project. This written analysis should also describe your greatest personal challenge over the course of the project and how you overcame it.
* **Cohesive written summary:** Describes how you contributed to each role that you didn’t play. This might have occurred via team discussions, peer reviews, or other means.

**Project and Team Summary (60 points)**

The project summary should be a two paragraph written analysis that summarizes the project, illustrates the teamwork of the group, and documents lessons learned. The summary should be written as if to be used in an interview or cover letter, and includes all of the following:

* Communication protocols used, including any challenges, how they were resolved, and what they would do differently next time
* Team strengths, including tips and tricks they would want to share with a new cohort kicking off the project
* Topic addressed
* Machine learning model used
* Results of the analysis

Segment 1, you'll work with your project team and begin assigning roles and tasks. You'll also choose the data that you'll use for the project.

What You’ll Accomplish

By the end of this segment, your team will have created the foundation for your final project. Defining the roles of your team members and establishing a communication structure will give you an excellent start. Your team will also complete the following tasks:

* Decide on a topic for the project—that is, think of a question that using data can answer.
* Create a repository for the project, and then invite the other team members to join it.
* Source a dataset or, if applicable, multiple datasets that will suit your needs.
* Begin to clean, organize, and perform an exploratory data analysis on your datasets so that they're ready for analysis.
* Include mock-ups of a machine learning model and a database.

Planning Your Schedule

Project weeks can be intense. So, make sure to plan extra time for your tasks. An excellent chance exists that you'll either find yourself researching issues you haven’t encountered before or want your code to do something more advanced than what the course has covered. That's all OK!

This section of the project is about exploring and building a foundation. So, feel free to start on a few ideas—but don’t get too attached to anything that you create this week. Which parts of a project will take too much time is sometimes hard to know until you start working on them. So, use the first part of this week to dive into the areas of the project that you feel the most unsure about. Take notes about the work you do, and leave time at the end of the week for a discussion. That discussion should focus on determining which ideas to discard and not just on which ideas to work on. Ask yourself the following questions: Which ideas are essential to the project? Which would be nice to have? Which would involve too much work to complete on time? If you can answer those questions before the start of the next segment, you’ll remain on track!

Before diving into your project work, you need to create a solid foundation to build on. This means familiarizing yourself with your role and responsibilities for the week. Your team also needs to establish communication strategies and prepare a repository.

This week, you need to think of a question that data can answer. And, you need to source the data that you intend to use. Then, during class, your team will meet with your instructor to discuss the topic of your project. Pay attention to the feedback that you get before moving on to the next segment! By the end of the meeting, you'll know what you'll work on for the remainder of the course.

Your team will also need to create a mock-up of a machine learning model and a mock-up of a database with either sample or fabricated data. By doing so, you'll gain a firmer grasp of how you want those pieces to interact.

Completing an exploratory data analysis will help you determine if your topic is feasible or if further consideration (or even a different topic) is needed. Doing so will also help you prepare for that meeting with your instructor. Thinking about the metrics that you'll include in the machine learning portion is advisable.

To sum up, your team needs to complete the following before meeting with your instructor:

* Decide on a topic, source the data, and perform an exploratory data analysis.
* Create a repository and establish the individual branches for the team members.
* Create a mock-up of a machine learning model.
* Create a mock-up of a database.
* Decide which technologies you’ll use.

## Working as a Team

Once your team has agreed on a topic, you should decide how to organize the team. Note that designating roles often proves useful in addition to assigning work task by task. This will be discussed more in depth later in this lesson.

All the segments of the final project will be graded in two parts: project and presentation. It can be tempting to avoid working on the presentation until the technical pieces work perfectly. But, don’t make that mistake! Keeping design notes, steadily adding work to your presentation as the work progresses, and regularly editing your presentation will make the difference between a rushed, mediocre slide deck and an engaging, portfolio-worthy showcase of your work.

**Project (60 points)**

In this segment, the project deliverable includes the following:

* A detailed README.md file (20 points)
* At least four commits per team member (20 points)
* A database that stores at least two tables (or collections) for the project (20 points)

We encourage you to choose and explore a dataset before committing to an idea for your project. Designing a database to hold your data will often reveal potential project ideas that you might not have thought of before. Try to spend the majority of this segment focused on digging into the data that you have and understanding any valuable insights that it might hold.

While working with the data, be sure to make regular commits. Each team member should have at least four commits by the end of the segment. And every time that the team meets, you should start and end the session by spending a few minutes reviewing and updating the README.md file. This file should always reflect the current status of the project, any specific needs or challenges that exist, and the plans for future work. Make sure to inclue links to your data, your slide deck, and any other important references. Don’t worry if the README.md file doesn’t yet appear worthy of showing in a job interview. This file will reflect the status of the project—so once the project is showcase ready, the README.md file will be, too.

**Presentation (40 points)**

In this segment, the presentation deliverable includes the following:

* The selected topic and the reasoning for that selection (20 points)
* A description of the data (20 points)
* The questions that the team plans to answer with the project (20 points)

During this early phase, focus on outlining and documenting your ideas, questions, and data. You don’t need to have a slide deck at this point. If the team feels more comfortable outlining this information in the README.md file, that’s perfectly acceptable. This kind of documentation and outline is critical not only for the final presentation but also to organize and align all the team members on the goals and deadlines of the project.

As a team, open a discussion about the topics that you'd like to research. Keep in mind that you'll need a sufficient source of data to analyze—with a minimum of 1,000 data points. In fact, the data is your only limitation. Lots of data sources exist, so feel free to take advantage of data resources like [KaggleLinks to an external site.](https://www.kaggle.com/datasets" \t "_blank), [data.worldLinks to an external site.](https://data.world/datasets/open-data" \t "_blank), or government sites. Web scraping also presents an option. Stretch your skills! Just make sure that you don't delay for too long when making a decision. You should spend the majority of your time on analyzing data and not on searching for or getting data.

Establishing a communication protocol is also important. For example, create direct messages for only team members in Slack and determine a daily meetup time for everyone. You might also include a communication protocol in case of an emergency. This might include whom to contact, whether to send an emergency message through Slack or email, and other details.

## Assigning Roles

Organizing the team is a critical first step to any project. But, no requirement exists for specific roles or team organization. Instead, each team can decide on a project management scheme that will work best for the members. Assigning roles is a popular solution. For example, assigning the project manager role to one team member can help to both keep the work on schedule and make sure that nothing gets missed. Assigning the QA engineer role to another member can make it easy for everyone to get their code reviewed. Other roles can include business analyst, software architect, database administrator, and many more. For more ideas, see [Software Project Team Roles and ResponsibilitiesLinks to an external site.](https://sherrierose.medium.com/software-project-team-roles-and-responsibilities-152a7d575759).

On a team of this size, everyone will have to take on more than one role at some point. But, it's advisable to establish early on how to distribute the largest responsibilities among the team. Throughout the project, make sure to regularly revisit those primary roles to find out if anyone either needs help or has extra time that can be used elsewhere. Then, as smaller tasks arise, it will be easier to figure out who’s best suited to tackle them.

After organizing the team from an overall perspective, it will be time to get specific and distribute some individual tasks. Let's review the requirements for Segment 1.

## Navigate Through Branches

Before you can create more branches, you need to switch back to the main branch. (Otherwise, you'll create a branch of a branch. Then, the work in the newest branch will need to be merged into the original branch—and things can quickly become confusing.)

To switch from a newly created branch back to the main branch, run the git checkout main command.

The git checkout <branchname> command also works for any branch. To switch to any branch, run git checkout <branchname>, where <branchname> is the name of the branch that you want. If that branch doesn’t exist, use git checkout -b <branchname> to create it.

The team should include an ERD of the database and a document that describes how the ERD both integrates into the database and works with the code. To test the database, the team needs to use either sample or fabricated data.

## Submitting the mock-up of the Database

By this week's virtual class, your team needs to submit the mock-up of the database. The team should also submit the ERD, the document describing the ERD, and the data used for testing. Make sure to push all the database-related work to its own branch in the repository. Then submit a link to the repository branch, as explained earlier in this lesson.

Consider the following questions: Which tools will best fit for your project? Which tools will you use for each section? (For example, you’ll use different tools for data cleaning and data storage.) How will you build the dashboard?

All the technologies will play a large part in the success of the project. So, knowing them in advance will help smooth the workflow. You don’t need to submit this list of technologies separately. But, you should document it in the README.md file so that all the team members can reference it as necessary