Phase 2 Project
Box Office Films

// FLATIRON SCHOOL

Agenda

- Project Prompt
- Project Deliverables
- Schedule

Project Prompt

Project Prompt

You are charged to explore the data to find what makes a movie successful

Provide 3 concrete recommendations to a “new” movie studio on what films to produce - support via statistical testing

Utilize Simple Linear Regression to quantify a numeric response variable relationship (revenue, ROI, budget, etc...)

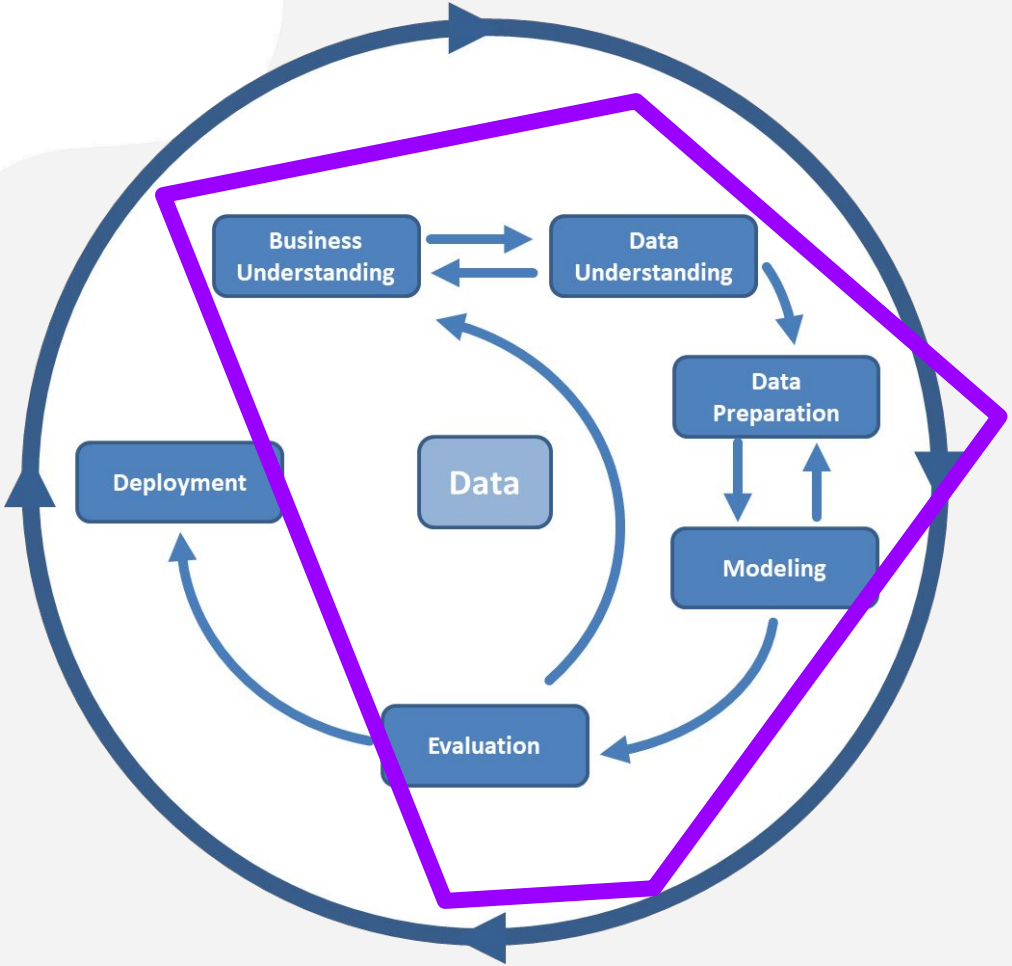
Project Deliverables

DS Process: CRISP-DM

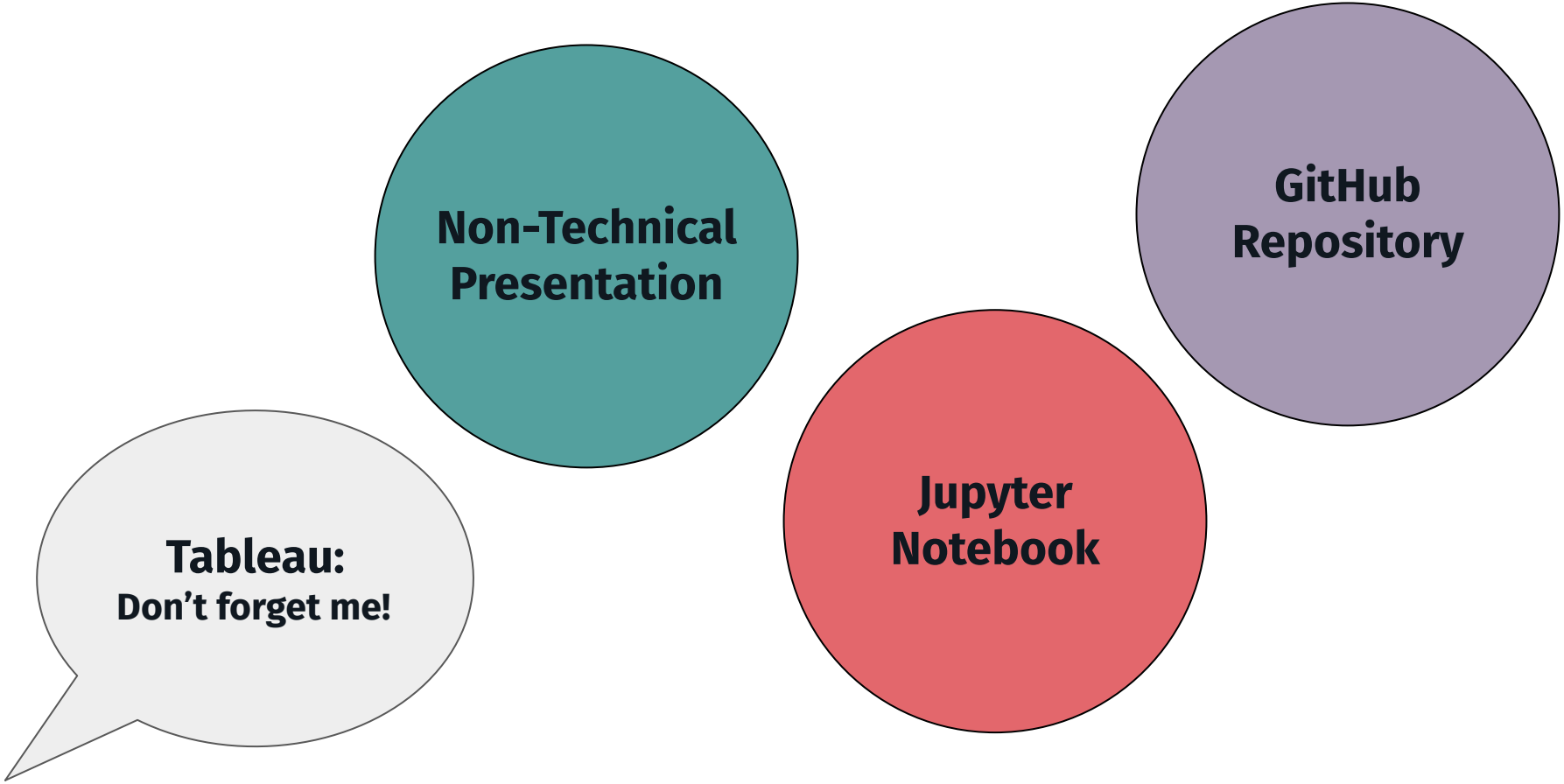
Consider the **CRISP-DM** process and headers while creating each deliverable.

Modeling:

1. **Statistical Tests**
2. **Simple Linear Regression**



Project Deliverables



Non-Technical Presentation

- Slide deck for a **five minute** presentation
- **Non-technical audience**
- Professional style
  - Light on text
  - Effective template
  - Legible and labeled visualizations

Example slide deck

Non-Technical Presentation

Tell a Story:

Beginning

- Overview
- Business Understanding
- Stakeholder
- Key Business Questions

Middle

- Data Understanding
- Visuals from EDA
- **Statistical Results (non-technically!)**

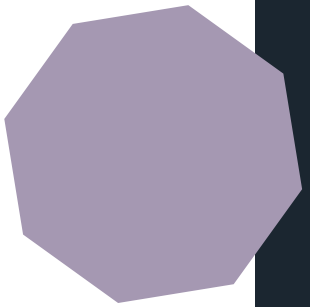
End

- Recommendations
- Next Steps
- Thank You Slide

GitHub Repository

- Where your project lives and grows - want to see a consistent commit history throughout
- **This will be part of your portfolio at the end of this course!**
- Recommend **starting your repository from scratch** rather than forking the template repository

Example repository and templates



GitHub Repository

Must-Haves

1. **README.md**
   - More detail on the next slide
2. **Commit History**
   - Commit history with clear messages
   - Contributions throughout the project period
3. **Organization**
   - Clear folder structure
   - Clear naming conventions for files and folders
   - Technical notebooks and presentation file are easily located
4. **Notebook**
   - Final technical notebook on main level of repo
   - Working notebooks (if applicable) in subfolders
5. **.gitignore**
   - Ignores large files as well as junk files
   - GitHub's python .gitignore template
   - **PRO TIP: Add the unzipped sql database file to your .gitignore immediately**

GitHub Repository

README Sections

Your README should act as a **high-level technical summary**

- **General Overview**
- **Business Understanding**
  - Include stakeholder and business questions
- **Data Understanding**
  - Source of data (either describe or link)
  - Description of data (high level, go into more detail in your technical notebook)
- **Modeling**
  - Describe techniques or methods
  - Written interpretation of results (final model)
- **Conclusion**
  - Summary of conclusions / recommendations
- **Repository File Structure**
  - (nice-to-have not need-to-have)

Jupyter Notebook

- Blends code, markdown, and visualizations to tell the **full story** of your project (content may overlap with your non-technical presentation and README)
- Includes **justifications and rationale** for every decision made throughout the project
- Notebook should be free of errors and run from top to bottom
- Use CRISP-DM steps as markdown headers to divide your final notebook into **sections**

Important Links

- **Project Description**
  - Explains the project goal, dataset, and deliverables
  - Contains rubric explanations

- **Checklist Overview**
  - Use to check off requirements

Working Groups and Schedule

Group 1:

- Jee Soo
- Elif
- Ricky

Group 2:

- Anbita
- Jayla

Group 3:

- Eli

Group Project Best Practices

1. Get to Know Your Group Members
2. Define Individual Project Contributions
3. Meet Regularly
4. Communicate Actively, Clearly, and Transparently

Schedule

Project Kickoff: Right now!

Check Ins: Tuesday AM/PM

Office Hours: Mon, Tues, Wed, Thurs

Thursday AM: Practice Presentations

Friday AM: Final Presentations (Time TBD)

Final Due Date

Friday 10/25 - 10:30 AM ET:

- Submit deliverables on Canvas!
  - Github link
  - PDF's

Steps to Get Started

Read Project Description

First Things First

1. Create github repo
2. Add group members as collaborators
3. Separate branches & notebooks

Plan Ahead

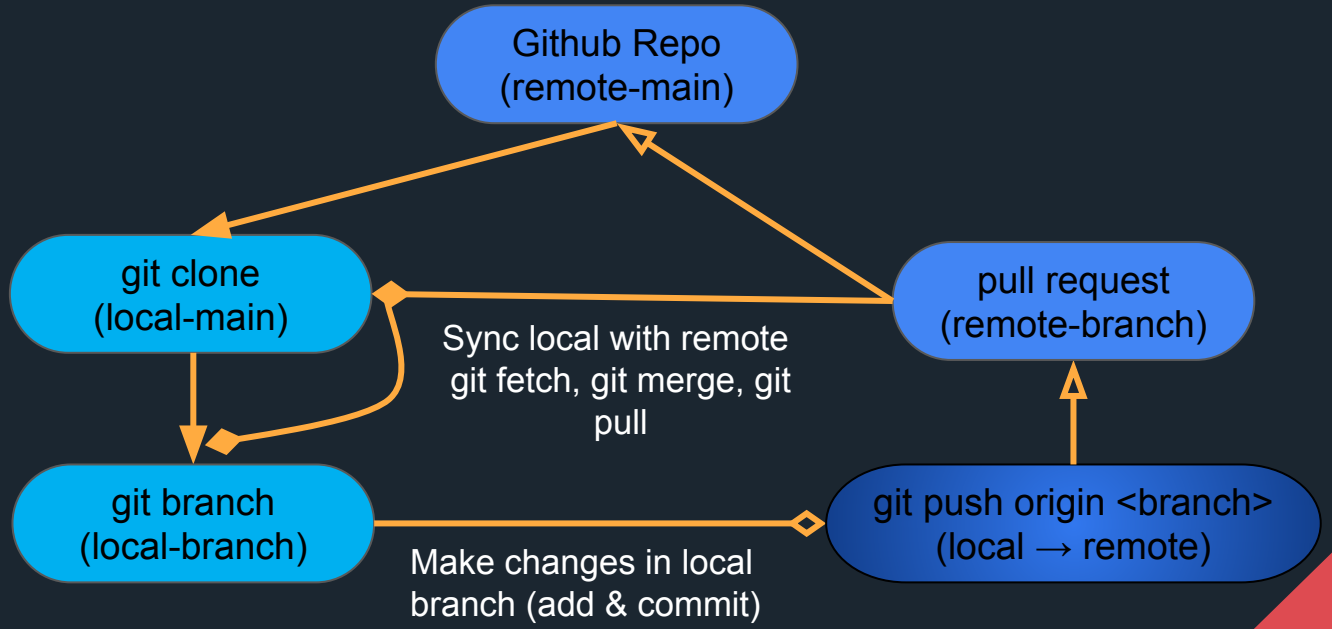
- Set group meet times
- Strategy for pull requests/git
- Utilize project 'board' if desired: Notion, Trello, or Monday

Think Data

- Master 'cleaned' dataset
- Might not be able to use all data files
- Joins, merges, etc
- Metric of Analysis

Gitflow for Branching Work

(Avoid merge conflicts)



Questions?