Module 7: Final Project

Get hyped. The next 3 weeks of your life are all about your final project! This is the capstone project that you'll be showing off to demonstrate all the things that you've learned over the last few months. This is awesome!

But, before we dive into what your project is going to be about we need to establish some ground rules.

Ground Rules

- Show up by 9:00 and leave no sooner than 6:00
- Participate in stand-ups, retros, and check-ins
- Finish acquiring your data and have a baseline model and dataset (**MVP**) by the established due date
- Meet with your scrum group in the morning and at the end of the day
- Adhere to the project decisions made with project managers (coaches)
- Communicate issues with your project managers (coaches)
- Be courteous to your classmates; Work together to solve problems
- Research and debug before escalating to your project manager
- Don't proceed beyond basic data exploration until you have pitched your project and it has been approved by your instructors.

Final Project

This is the time to dive into the data science process using techniques we've learned. This is the time to build the model that you've always wanted to but never could, and really put your skills on display.

Because we're asking you to show off a specific set of skills, we have some requirements. It should be obvious that one of the requirements is that you need to implement the knowledge that you learned while at Flatiron. This isn't the time to build a new classification algorithm or do reinforcement learning. You've done a ton of learning already - it's time to apply all of that knowledge.

Project Requirements

The listed requirements below are guidelines that should help you to determine what the complexity of your project should be. They are not hard and fast rules. Final project approval is up to your coaches, who will be acting as project managers.

Your final project is an elevated end-of-mod project . It can be from any of the topics that we've covered over the past 12 weeks and should demonstrate an understanding of data science concepts.

Things You Can Build

Here are some suggestions of things you can do; you're certainly not limited to these!

- Regression models (linear, CART, etc)
- Classification models (KNN, CART, logistic, etc)
- Time series models
- Clustering
- Neural Networks
- Recommendation engine

Data Science Concepts/Components You Can Include

- Databases (SQL, MongoDB, etc)
- API Interaction
- Hadoop/Spark components
- Natural Language Processing
- Image Processing
- A webapp to showcase your project

What Do We Want?

The project rubric is at the end of this document, but in summary:

1. A **clean Github**! Try to make it as reader-friendly as possible. How would you organize your project repo to best demonstrate your project to a recruiter?
    a. Readme, clean notebook, gitignore, etc.

2. Within your Github, we really want you to **demonstrate the data science process** in your code/notebooks. As we've learned over the past 12 weeks, the process can be broken down into:
    a. Identify the problem
    b. Data wrangling/cleaning
    c. Feature selection/engineering
    d. Model building/tuning
    e. Evaluation
    f. Final product

3. Your project should have a **narrative** that ties the whole thing together. *Why are you doing your project?* At each step of the way you should be able to justify your decisions in relation to the purpose of your project. It's very important to contextualize!

4. It seems like common sense, but it's very important that you **understand the models and the concepts** that you're incorporating into your project. If you use Singular Value Decomposition in your recommendation system, you should be able to answer the question: How does SVD work? These project-related questions are guaranteed to come up in job interviews! As a **bonus**, be able to explain these concepts at different levels. i.e. *explaining XGBoost to a technical/non-technical person*.

5. A **final presentation**. On Day 5 of Week 3, we'll be doing project presentations. This is the time for you to present your hard work to your peers! Presentations should be within 4.5 minutes long. There will be a **hard stop at 5 minutes**, including questions!

Roles

Project managers - Coaches and Instructors, provide project approval and guidance

Scrum group - Students will be assigned groups of 5-6. All group members will become intimately familiar with each others' projects. Responsibility of scrum group members are:

- Morning stand ups
- Provide each other with advice, brainstorming, and resources
- End of day stand downs and a final retrospective
- Weekly post mortems

Daily Check-In

1. **Intro:** My name is _____ and I'm a data scientist with a background in _____.
2. **Elevator Pitch:** The world has a problem: _____. I'm solving it by doing _____. (30s)
3. **Accomplishments:** I've completed _____ since our last check-in.
4. **Plans for today:** Today I will plan to accomplish _____.
5. **Blocking Issues:** I need help with _____.

Schedule

For much of Mod 7, your time is your own. There will be a few lectures and other activities, but most of the time will be dedicated to your project. This does ***not*** mean you get to come in late or leave early or take four hour lunch breaks. Think of this as a work environment. Your coaches have sent a detailed schedule; below are some of the particularly important dates.

Before Project Start

Nov 11 - Project Ideas

Submit 3 written projects ideas by 10PM. You're expected to have:

- the real world problem you want to solve
- potential data sources

Nov 13 - Coach check-in

Check in with coaches to discuss your submitted project ideas and choose the best one. Receive proposal worksheet

Nov 15 - Project proposals

Submit proposal worksheet (See the project proposal suggestions later in this document)

Week 1

Nov 20 / Day 1 - Coach check-ins
Morning check-in with coaches to make sure everything is on the right track with a timeline for your project going forward

Nov 21 / Day 2

Nov 22 / Day 3 - FSM, Coach check-ins
First Stinky Model - You should have cleaned data to the point where you can run your first model. End of day mandatory check-in with your assigned project manager

Week 2

Nov 25 / Day 1

Nov 26 / Day 2 - Coach check-ins
Mandatory check-in to go over the technical aspects of your project

Nov 27 / Day 3 - Minimum viable Product

You should have a Minimum viable Product. How can you model be used to benefit society? It is not enough to build a model on clean data, how will it be useful and in what scenarios will it be used?

Week 3

Dec 2 / Day 1 - Substantial Completion, Practice Presentation

By week three, day one, you should be done with core modeling and data wrangling. You should have your project done. Remaining time to work before science fair, should be cleaning notebooks, the readme, and working on the presentation.

Dec 3 / Day 2 - Data Science Projects Complete, Document Review, Practice Presentations, & Feedback

Dec 4 / Day 3 - Documentation Review & Feedback

On day 3, you should have all documentation complete, everything done. This is the time for coaches and instructors to provide meaningful feedback on how to best make your online presence shine.

Dec 5 / Day 4 - Graduation & Data Science Project Expo

This is the day that you'll be showing off your projects to the rest of the school, friends and family, and prospective employers. Graduation will be from 5-5:30, here on campus, and **guests are invited!** Please check with your instructors to make sure we account for enough space.

HARD STOP at 2:00 PM*

Science fair lasts from around 5:30 to around 7:30, so prepare for a long day of talking to people about your project. People do get job interviews out of science fair, so make sure to put your best foot forward, dress a little nicer than you normally would, and get that elevator pitch ready! **The elevator pitch for the science fair should be a short overview of your project around 3 minutes, not a 15 minute presentation on how hard it is to train a specific model.**

Really think about the main aspects of your project you want your audience to take away.

What did you build?
Why did you build it?
How does it contribute to society?

Often, unless you did something revolutionary, people won't care about your data cleaning process. (Sad. I know.) Imagine you want to tell your non-technical friends about your project. Can you do that in under 3 minutes? (Of course, be prepared in case you're approached by someone who does have technical questions.)

*Time may change

Dec 6 / Day 5 - Breakfast, project presentations & retro

You made it! Start your morning with a nice breakfast, show off your projects to your classmates, and participate in a small-group retrospective (often called a "postmortem"). This will be the time to talk about what went right, what went wrong, what roadblocks you're still up against and your plan of attack for getting over them.
You can expect to be done with the day by 3:00 pm.

Helpful Tools

Kanban/Scrum Board

Just because you're working solo doesn't mean you don't need to stay organized. In fact, because this will be the most complex project you've made at Flatiron, you'll *need* something to keep you organized. We recommend Trello or a Github Project Board. Use this to track what you're doing and what you need to work on. It's also a great idea to keep track of bugs that you're not going to immediately fix.

Pomdoro Timer

If you don't take breaks, you'll end up hurting your eyes, getting an RSI or burning yourself out. The Pomdoro Timer method lets you put in solid chunks of work while also giving you regular breaks. We like Marinara Timer, since it's nicely customizable.

Project Resources

Postman - Test API calls

Heroku - Simple, free web hosting for flask or django in python

DB Browser - SQLite interface for making calls to a database

AWS - Amazon Web Services for running models on the cloud

Proposal Questions:

Question:

- What problem are you trying to solve?
- What is the industry or realm this applies to?
- Who would be your target audience?
- What personal connection do you have to this topic?
- What would the minimum viable product involve?

Business Understanding:

- What pre-existing research or papers in this field can you reference?
- Has anyone else worked on this specific problem/dataset? If so, how will your work build on theirs?
- What terms or concepts will you need to research for this field?
- What impact would your answer have on the real world if your analysis were put into production?

Data:

- What's your data?
    - do you have a pre-made dataset?
    - Is your dataset scraped/or use an API?
    - Collect tweets?
- What format will your raw data be in?
- Please attach a sample of your data, evidence that you can scrape or call an api.
- What will you need to clean? Have you already cleaned it?
- What are the dimensions of your dataset?

Tools:

- What python libraries will you need to gather, clean, explore, and model your data?
- Where will you be performing your analysis? On your machine? In the cloud?

Methodologies:

- What type of problem is this? Classification? Regression? Unsupervised? Supervised?
- What modeling algorithms do you plan to use?
- If you answered “NN”, what are two other algorithms you could also use?
- What methodology will you use for your **baseline model**?

Work Schedule:

- Please provide a work schedule with clear work phases and deadlines

Project Rubric

This is your technical resume and our brand. Achieving this level of quality will make your stand out from the rest of the market.

General Expectations

Area	Successful Project	Incomplete Project
Documentation	Project has a README file clearly documenting each step in the CRISP-DM workflow.	The README file is missing or incomplete, and/or some steps in the CRISP-DM workflow are not documented fully.
Reproducible Science	Any Flatiron School DS graduate could reproduce your work by following the simple instructions found in the project's README file.	Instructions to reproduce the work are incomplete or confusing, or manual steps are required to prepare data.
Copy Editing	The README file and all other documentation are well written with proper spelling and grammar.	One or more obvious typos, spelling errors, and/or grammar errors exist in the README file or elsewhere in the documentation.
Jupyter Notebooks	Jupyter Notebooks are focused on EDA, visualization, and presentation. Custom functions and classes are imported from Python modules and are not created directly in the notebook.	Some or all of the Python code needed to reproduce the work is contained directly in Jupyter Notebooks.
File Naming	File names follow a consistent naming convention. Recommended: all lowercase, no spaces (e.g. "food_prep.py" and not "Food Prep.py"). File names are not used for version control.	File names follow an inconsistent naming convention, or file names contain words like "new" or "final" or "v2".

Version Control

Area	Successful Project	Incomplete Project
Project Organization	All project code is contained in a GitHub repo with separate subdirectories for source code and data.	The project repo is incomplete, or the organization of the repo is confusing.
Git Commits	The project shows a steady commit history, on the order of **2-3 commits per day** during the capstone project period.	The commit history is sporadic or inconsistent, and/or large portions of work are contained within a few monolithic commits.
Commit Messages	Each commit message begins with a **verb in the present tense and imperative mood**. Each commit message describes the high-level intent of a specific change.	Commit messages are vague, are written in an unprofessional style, and/or do not describe the high-level intent of the changes made.
Repository Name	The repo name is creative and original and/or describes what the project does to solve a problem.	The word "Capstone" or "Project" appears within the repository name, and/or the name is not appropriate to the nature of the project.
.gitignore File	The project contains a **.gitignore** file based on GitHub's default .gitignore file for Python projects, excluding items such as .ipynb_checkpoints/, __pycache__/, etc.	There is no .gitignore file, and/or the repo contains unnecessary directories such as __pycache__/, .ipynb_checkpoints/ and files such as .DS_Store.
Branching	The final project lives on the master branch of the repo. Other branches, such as unmerged feature branches or historical checkpoints, may exist.	The final project is stored in one or more branches other than the master branch, or it is necessary to view multiple branches to see the entire project.

Code Quality

Area	Successful Project	Incomplete Project
Style	All Python code complies with an industry standard, e.g. Google Style, Numpy Style, or Black.	Python code style is inconsistent or incorrect.
Docstrings	Each Python module, function, and/or class contains a valid docstring containing a mandatory one-line description and, optionally, a more detailed explanation of the code. All docstrings are current and accurate.	Some or all docstrings are missing, incomplete, misleading, and/or out of date. Or, comments are used in place of docstrings.
Naming Style	Each object name follows PEP8 conventions (i.e. classes are written in CamelCase and functions are written in snake_case)	The naming of objects is inconsistent and/or does not comply with PEP8.
Variable Naming	Variable names represent what is contained in the variable, not its type. Single objects are singular nouns in snake_case, or ALL_CAPS for constants. Collections containing multiple items are plural nouns.	Variables are named after the type of the object (e.g. `lst`) or follow invalid naming conventions (`NewDF`).
Function Naming	Function names begin with a present-tense, imperative-mood verb, e.g. `clean_data`.	Function names use confusing naming style, e.g. `data_cleaner` or `MyFunction` or `f`.
Code Organization	Code is factored into functions that each do only one thing, with a typical length of 10 or fewer lines of code per function.	Code contains one or more "monolithic" functions that do more than one thing, and/or are longer than 10 lines of code.
Bonus: Class creation	Functions are bundled into a new class object that can be reused throughout analysis	

Slides

Area	Successful Project	Incomplete Project
Text vs. Graphics	A typical slide contains seven words or fewer. The main points are made visually using charts or graphics.	Slides contain numerous words and/or detailed tables that distract from the student's presentation.
Screen Real Estate	The bottom third of the screen is reserved for titles or unimportant graphics.	Important content is displayed on the bottom ⅓ of the screen, where it may be blocked by heads etc.
Sourcing	When applicable, your slides cite the origin of other people's images, graphics, etc.	Slides contain plagiarized or uncited content.
Sharing	Slides are available for anyone to view (i.e. Google Slides public link or available as presentation.pdf on GitHub). The location of the presentation is specified in the README.md file.	The presentation materials are incomplete or are not publicly available.
Topics	Slides cover, at minimum: introduction, business understanding, data understanding, data preparation, modeling, evaluation, deployment/demo, and next steps for future improvements. The final slide includes the graduate's contact information.	One or more key topics is missing from the presentation.

Business Understanding

Area	Successful Project	Incomplete Project
Problem Definition	The README file clearly explains the problem that the project sets out to solve.	The project has vague or unclear goals, e.g. to explore a dataset without a specific problem in mind.
Success/Evaluation Criteria	Criteria for success and/or evaluation are laid out in the Business Understanding section of the README.	The README file does not include a description of project evaluation criteria.

Data Understanding

Area	Successful Project	Incomplete Project
Public Data	README.md file contains a hyperlink to where the public data can be found, with an explanation of the data source.	README.md fails to contain information on where someone can get the public data or who shared the public data.
Confidential Data	README.md file explains that this project used confidential data and will not be made publicly available. In the event that confidential data is not available, fake or anonymized is included as a substitute.	README.md fails to explain the nature of the confidential data source.
Example Data	If a public or scraped dataset is used, the repo contains the dataset, if it is sufficiently small. If the dataset is too large to include in the repo, a shard of data is included that could be used to replicate the data pipeline (potentially with reduced model performance) without downloading or scraping additional data.	No example data is provided.

Data preparation

Area	Successful Project	Incomplete Project
Data Preparation	Original raw data is preserved. All data preparation steps are reproducible and are documented *in code*.	Raw data was manually altered, or some data preparation steps are not documented in code.
Data Pipeline	The data cleaning and processing steps are contained within functions and reproducible. All models and transformers are fit only on the training data, not on the entire dataset.	The data cleaning and processing steps are not contained within functions and not reproducible, or transformers are fit on the entire dataset and not just the training data.
Cross Validation	Cross validation is used correctly to evaluate model performance.	Cross validation is not used to evaluate model performance, or cross validation is used incorrectly.
Data Leakage	The model is not trained on features that leak the target variable. If the training data is collected over time, the train/test split is (most recent data is held out) instead of random.	Did not exclude features that leaked information during model selection and hyperparameter tuning.

Modeling

Area	Successful Project	Incomplete Project
Python Files	Functions are contained in .py files which are then imported in Jupyter Notebooks.	All functions are stored in Jupyter Notebooks.
Documentation	Whenever possible, documentation (README.md, comments, Markdown cells) are used to explain *why* modeling decisions are being made. Another data scientist engaging with the project can understand the context of technical decisions that were made during the modeling process.	The only form of documentation is in-line comments or is non-existent all together.
Baseline	During model selection, do the simplest model first (i.e. guessing the majority class, Logistic Regression, NaiveBayes, Linear Regression) before trying more complex and less interpretable models (i.e. neural networks, random forest)	Did not have the simplest model as a baseline to compare their final model against (i.e. no sense of how well the final model does against something simpler)

Deployment

Area	Successful Project	Incomplete Project
Relationship to Business Understanding	The data product directly answers the business question and addresses the specified problem.	There is no final data product, or the data product is not relevant to the stated business question.
Web Presence	The bare minimum here is a Github repository. You can extend this with a Medium post, a Github.io homepage, or for the ambitious, a web app (like Flask) that is deployed to the internet via AWS, Heroku, or other.	The project has no presence on the Web.

Next Steps

Area	Successful Project	Incomplete Project
Model Improvement	Project "next steps" include potential ideas to improve the model through feature engineering, parameter tuning, etc.	No reflection is given as to ideas for how the model could be improved.
Product Roadmap	Project "next steps" include ideas for future product improvements that further address the original business problem.	The project does not include ideas for future improvements.